TESI DI DOTTORATO

UNIVERSITA' DEGLI STUDI DI NAPOLI "FEDERICO II"

DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELLE TECNOLOGIE DELL'INFORMAZIONE

DOTTORATO DI RICERCA IN INGEGNERIA INFORMATICA E AUTOMATICA

An insight in Cloud Computing solutions for intensive Processing of Remote Sensing Data

Claudio De Luca

PHD Coordinator:
Franco Garofalo

Tutors:
Valentina Casola
Francesco Casu

Academic Year 2016 – 2017

Index

Introduction

The investigation of Earth's surface deformation phenomena provides critical insights into several processes of great interest for science and society, especially from the perspective of understanding the Earth System and the impact of the human activities. Indeed, the study of ground deformation phenomena can be helpful for the comprehension of the geophysical dynamics dominating natural hazards such as earthquakes, volcanoes and landslides. Furthermore, such information can be very useful also to detect, monitor and limit the effects of anthropogenic actions, such as ground-water exploitation, oil and gas extraction, gas capture and storage, mining activities, tunnelling, building and management of dams and flood defenses, and transportation.

In this context, the microwave space-borne Earth Observation (EO) techniques represent very powerful instruments for the ground deformation estimation, especially thanks to their characteristics of large spatial coverage and cost effectiveness. In particular, the Differential Synthetic Aperture Radar Interferometry (DInSAR) is regarded as one of the key techniques for its ability to investigate surface deformation affecting large areas of the Earth with a centimeter to millimeter accuracy. Among several DInSAR algorithms, a well-known approach is the one referred to as Small BAseline Subset (SBAS), which allows the computation of displacement time-series and corresponding mean deformation velocity maps in different scenarios (both natural and man-made).

The SBAS algorithm, and more generally the advanced DInSAR techniques, found their success on the large availability of data archives acquired over time by several satellite systems. Indeed, the current radar EO scenario takes advantage of the widely diffused long-term C-band European (e.g., ERS-1, ERS-2, and ENVISAT) and Canadian (RADARSAT-1/2) data archives, which have been acquired during the last 25 years, as well as of data sequences provided by the X-band generation sensors, such as the COSMO-SkyMed (CSK) and TerraSAR-X (TSX) constellations. Moreover, a massive and ever

increasing data flow is further supplied by the recently launched (April 2014) Copernicus (European Union) Sentinel-1A satellite, which is characterized by a wide ground coverage (250 km of swath) and a revisit time of 12 days that will reduce to 6 days when the twin system Sentinel-1B will be placed in orbit on April 2016. Such characteristics (ground coverage and revisit time) together with the global coverage acquisition policy will lead to an exponential increase of the provided data flow that, after the launch of Sentinel-1B, will reach about 6TB per day. This acquisition rate is a good indicator of the amount of data that has to be faced by the EO community. To better stress this aspect, we can consider that the ERS-1/2 sensors in about 20 operative years (from 1992 to 2011) collected approximately 540 TB of data that, with the Sentinel-1 acquisition rate, are reached in only 6 months. Same considerations could be done for the CSK constellation that acquires about 600 GBytes of data every day and TSX that reachs 90 GBytes of data per day.

This huge amount of data is of extreme benefit for the advances of scientific research in the EO field. However, the data management, storage and processing is becoming a critical issue. Indeed, the in-house computing infrastructures, due to their intrinsic limitation for what concerns scalability, customization and cost, can hamper the effective exploitation of the above-mentioned EO data archives. Therefore, the shift towards computing technologies able to effectively process such an amount of data in reduced time frames, and the development of EO algorithms capable to exploit such computing infrastructures are nowadays of high priority.

In this scenario, a key role can be played by the Cloud Computing (CC) technologies, which can provide a theoretically “unlimited” amount of resources. The use of CC environments can overcome a lot of limitations of in-house solutions. Indeed, CC can be extremely helpful for resources optimization and scaling, performance improvements, complete customization of the computing architecture, on-demand employment of computing facilities, etc. Moreover, the increasing diffusion of public cloud processing services at very reduced costs is also pushing towards the adoption of such technologies in

scientific contexts. Indeed, recent initiatives of some cloud providers (such as Amazon and Google) are already moving in this direction. For istance Amazone Web Services (AWS) is making available through the Amazon Simple Storage Service (S3) the Landsat and Sentinel-2 satellite data that can be processed by exploiting the AWS Elastic Compute Cloud (EC2) facilities.

Such aspect points out, as another issue related to the huge EO data era, that it is becoming more and more crucial to move processors and tools close to the satellite data archives, being not efficient anymore the approach of downloading and processing data within in-house computing facilities, as usually done during the last 25 years in the EO field. In this sense, ESA recently funded the development of the Thematic Exploitation Platforms (TEP). TEPs are ESA ground segment activities aimed at demonstrating the benefit of new IT technologies for large scale processing of EO data. These initiatives encompasse on-demand processing for specific user needs, systematic processing to address common scientific needs, and integration of newly developed processors for scientists and other expert users.

This thesis work provides its contribution in the depicted rapid evolving scenario, with particular reference to the exploitation of cloud computing platforms within EO field.

Consequently, we first concentrate on the development of new effective algorithms for DInSAR data processing capable to benefit from the availability of high performance computing facilities. In particular, we develop a parallel computing solution for the SBAS processing chain (P-SBAS) that permits to generate, in an automatic and unsupervised manner, advanced DInSAR products by taking advantage of the employment of in-house architectures. Second, the proposed P-SBAS solution has been thoroughly evaluated in both in-house and CC infrastructures, through real case studies, by performing scalability tests and assessment of costs. Such studies also permitted to highlight the main bottlenecks of the P-SBAS implementation, allowing us to propose a novel architectural implementation to further improve the algorithm performances, especially in a CC environment.

Finally, the developed P-SBAS algorithm has also been integrated within the ESA GEP (Geohazards Exploitation Platform), with the main aim to provide an automatic on-demand web tool for the generation of accurate deformation time-series by exploiting ESA and Copernicus data. The availability of such tool for the EO community, which allows scientific users (not necessarily expert on DInSAR processing) to generate in unsupervised way and in unprecedent short time ground deformation maps of large areas, will open new intriguing and unexpected perspectives in the EO research field.

Specifically, this thesis is organized as follows:

- Chapter 1 – A brief introduction concerning the basic principles of SAR image formation is provided. Then, the DInSAR technique is presented, with particular focus on the SBAS algorithm that is aimed at the analysis of the temporal evolution of the ground deformation.
- Chapter 2 – We describe the parallel and automatic computing solution developed for the processing chain implementing the SBAS technique. The proposed parallel algorithm (P-SBAS) is based on a dual-level (multi-nodes and multi-cores) parallelization approach. An experimental analysis, aimed at quantitatively evaluating the computational efficiency of the implemented parallel solution through appropriate metrics, is also provided.
- Chapter 3 – We present an assessment of the P-SBAS algorithm within Cloud Computing (CC) environment. In particular, we investigate the scalable performances of the P-SBAS algorithm fixing the data-set and varying the number of exploited processors. To perform such an analysis we exploit the public cloud platform of AWS. A comparative analysis with an in–house HPC cluster, dedicated to the P-SBAS application, is performed. In addition, a cost analysis of the developed cloud architecture is presented.
- Chapter 4 – The analysis performed in Chapter 3 allows us to identify the major bottlenecks that can hamper the P-SBAS performances within a CC environment, in the perspective of

processing very huge satellite data-sets. So we developed a novel CC implementation of the P-SBAS algorithm, aimed at minimizing the data transfer and the network load, thus further improving the P-SBAS efficiency and scalability within CC environment.

- Chapter 5 – By benefiting from the work carried out in the previous sections, we integrated the P-SBAS within the ESA GEP to develop a web tool aimed at generating in an unsupervised way ground deformation time-series. Such a tool exploits the ESA's Grid Processing On Demand (G-POD) infrastructure, a part of the GEP platform. The direct access from the G-POD environment to the ESA data archives, together with a user-friendly interface, allows us to set up an efficient and on–demand P-SBAS processing web tool addressed to scientists.

Chapter 1

SAR Interferometry

Synthetic Aperture Radar (SAR) is a coherent active microwave remote sensing system, whose capability to effectively map the scattering properties of the Earth's surface has been already intensively investigated.

A SAR sensor, which can be mounted on-board to an aircraft and/or a satellite, has a side-looking illumination direction and is able to perform accurate distance measurements between the moving platform and the surface. Both the acquisition geometry and the physical characteristics of the scene contribute to the formation of the received backscattered radar signal (echo) which, properly processed, leads to the reconstruction of a complex high resolution SAR image. Moreover, being an active imaging sensor, it does not need of an external energy source to work and, exploiting of the microwave region of the electromagnetic spectrum, can be effectively used to detect areas affected by a significant clouds' cover. As a consequence of its flexibility, SAR technology mostly improved during the last 20 years and further techniques have been also developed, thus helping the scientific community on the interpretation of several geophysical phenomena.

One of the major applications of the SAR technology is represented by the SAR interferometry (InSAR) technique which exploits, in its basic form, the phase difference of (at least) two complex-valued SAR images (acquired from different orbit positions and at different times) to measure several quantities, such as topography, deformation, etc... This chapter, in particular, will address the key topics of the SAR technology, with a particular emphasis on the InSAR applications.

1.1 SAR History

Since the 1950's when Carl Wiley [1] made his first observations about the "Doppler beam-sharpening" phenomenon, SAR designs and related applications have grown exponentially.

In 1974 an alliance between JPL (Jet Propulsion Laboratory) engineers with a group of international ocean scientists led the National Oceanic and Atmospheric Administration (NOAA) to determine if an ocean application satellite, featuring a space-based SAR, could be achieved. Their efforts were directed to the realization of a satellite SAR system, which was launched in 1978 onboard to SEASAT [2].

SEASAT operated successfully from the late June to early October 1978 and, although explicitly designed to observe the ocean, was the first Earth observation satellite to provide SAR data suitable for interferometry. The interferometric effectiveness of the SEASAT SAR data for topographic mapping was, in fact, demonstrated eight years later by Zebker & Goldstein [3] and for detection and mapping of small elevation changes by Gabriel et al. [4]. It was followed by the Shuttle Imaging Radar-A (SIR-A) and SIR-B [5] flown in 1981 and 1984, respectively. Both radars were variations on the SEASAT radar, operating at L-band and HH polarization.

The 1990's witnessed a significant expansion of SAR missions with the launch of five Earth-oriented SAR satellites, as well as the pioneering interplanetary use of the Magellan SAR [6] to map Venus. The NASA continued with the Space Shuttle Imaging Radar SAR mission in 1994. The novel radar system was named SIR-C [7] and flown, in April and October, onboard to Endeavor. The SAR sensor, which operated simultaneously at three different frequencies (C-band, L-band and X-band) alternatively transmitting and receiving at both horizontal and vertical polarization, was developed by JPL, DLR (German Aerospace Center) and ASI (Italian Space Agency). Nevertheless, the real breakthrough in SAR interferometry has been achieved by the twin European ERS-1/2 (European Remote Sensing Satellite) sensors [8], launched in 1991 and 1995, respectively, which, with respect to the stability,

calibration, etc., represent nowadays the best goal achieved with SAR in the interferometry area. Moreover, a TANDEM mission, during which the two sensors operated in parallel, with ERS-2 followed ERS-1 on the same orbit with only one day of delay, was accomplished. The generated TANDEM interferograms presented an excellent phase quality and, therefore, very precise topographic maps were produced.

Also Japan and Canada gained a SAR system in 1992 (JERS) and in 1995 (Radarsat), respectively, but asset problems limited the real exploitation of these sensors.

The C- and X-band portions of the SIR-C radar were again flown in 2002 for the Shuttle Radar Topography Mission (SRTM) [9]. During this flight, a second receiving antenna was placed at the end of a 60 m mast, extended perpendicular to the main radar antenna. The purpose of the mast antenna was to provide a second receiving point in space for each radar pulse. The slight variations in phase will be processed into a height measurement of the reflecting point on Earth's land surface. During the 10- day flight, the space shuttle imaged the Earth between 60 degrees north and south latitude and a high quality, high resolution Digital Elevation Model (DEM) could be computed.

In practice, for more than a decade SAR images from space have been available on a reliable routine basis for scientific, public, and commercial users, independent of weather and daylight conditions. The most successful SAR workhorses have been ERS-1/2, JERS [10], Radarsat-1, and, since March 2002, ENVISAT [11]. Although the design of the ERS radars was guided by oceanographic questions, soon after the launch, interferometric (InSAR) applications took over. Generation of digital elevation models (DEMs), measurement of glacier flows, and mapping of earthquakes, volcanoes and subsidence are the most prominent fields of interest today. The majority of interferometric applications today uses data of ERS-1/2 and RADARSAT (both C-band SARs) while the only source of L-band data was the late Japanese JERS. Besides, a new generation of SAR satellites is now approaching: the German SARLupe and TerraSAR-X, the Canadian Radarsat-2, the Italian COSMO-Skymed and the Japanese Advanced

Land Observing Satellite (ALOS) [12]. They embody a lot of novel features, such as higher resolution, polarimetry, splittable antennas, and spot-light imaging mode. In this context, a key role could be played in the future, in particular, by the ALOS-PALSAR system (launched in January, 24th 2006), which should complement existing satellites as well as permitting InSAR over vegetated or other rapidly changing landscapes.

The more modern systems present a variety of imaging modes, which offer the possibility to expand the present-day interferometric data archive, by exploiting both high resolution and wide swath ScanSAR and TOPS [13] modes. Clearly, the possibility of the ScanSAR and in particular TOPS interferometry to generate wide- swath products in a single shot is highly attractive.

1.2 SAR Principles

A radar self-illuminates an area on the ground by transmitting a series of electromagnetic pulses and, after an accurate measure of the time delay between the transmitted and the received echoes, is able to figure out the distance (called slant range) between the sensor position along its flight direction (azimuth) and the illuminated targets on the ground. A radar system is well-characterized by its attainable spatial resolution, which measures the ability to distinguish two properly-separated objects. More precisely, if the objects are sufficiently separated, each will be located in a different resolution cell and will be discernible but, if not, the radar return will be a complex combination of the reflected energy from the two objects.

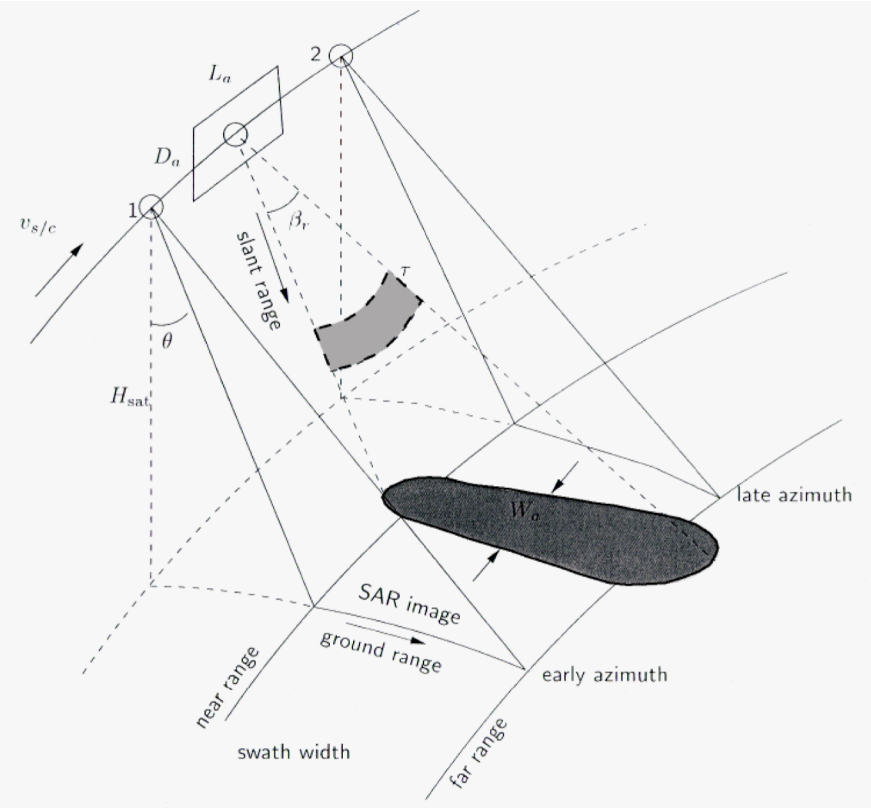


Figure 1.1: The configuration of a side-looking Real Aperture Radar (RAR).

With regard to the range spatial resolution, we can observe that two objects δr-away can be effectively discriminated if the received pulses (of width τ), are completely separated, only. Since δr corresponds to a time difference $\Delta t = 2\delta r\, c$ (where c is the speed of the light), the previous condition implies that

$$\delta r \geq \Delta r_{SR} = \frac{c\,\tau}{2} \tag{1}$$

herein Δr_{SR} is the theoretical slant range resolution, which can be also expressed, in a more generalized way, as a function of the corresponding pulse bandwidth $\left(\Delta f \simeq {}^{1}/_{\tau}\right)$

$$\Delta r_{SR} = \frac{c}{2\,\Delta f} \tag{2}$$

Moreover, it is usually preferred to refer to the ground range resolution, defined as the minimal distance on the ground that allows recognizing two distinct objects, which can be in turn obtained as the projection of the slant resolution (2) on the ground (see Figure 1.1)

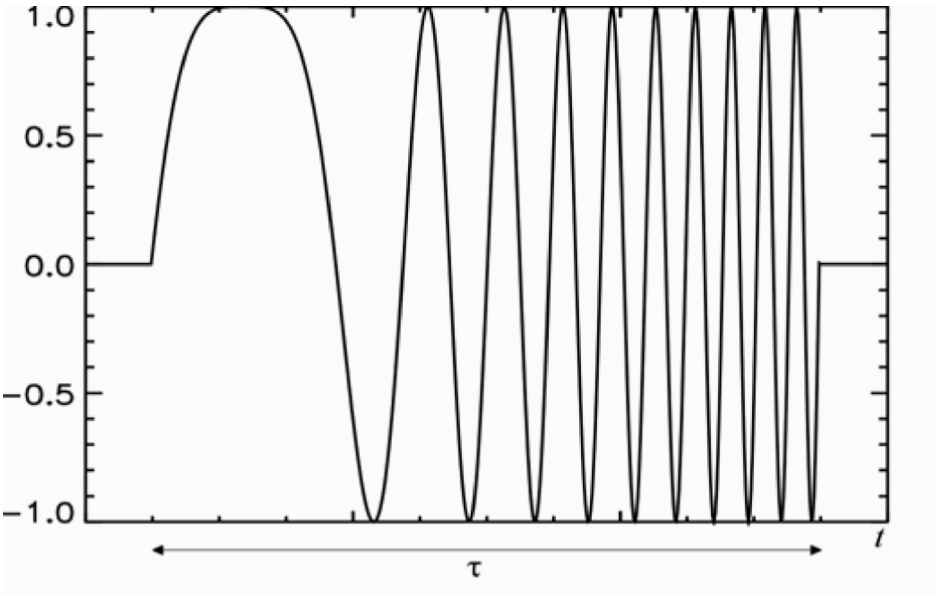


Figure 1.2: Chirp waveform (α >> 0).

$$\Delta r_{GR} = \frac{c}{2\,\Delta f\,\sin(\theta)} \tag{3}$$

wherein θ is the radar look angle. To give an example, if the ERS-1/2 parameters collected in Table 1.1 were considered, a corresponding ground range resolution of about 25 m would be retrieved.

The equation (1) suggests that finer resolutions can be achieved by using very short duration pulses only, thus requiring high peak power for a prescribed mean power operation. To overcome this

problem, modern radar systems (including SARs, also) transmit long linear frequency modulated pulses

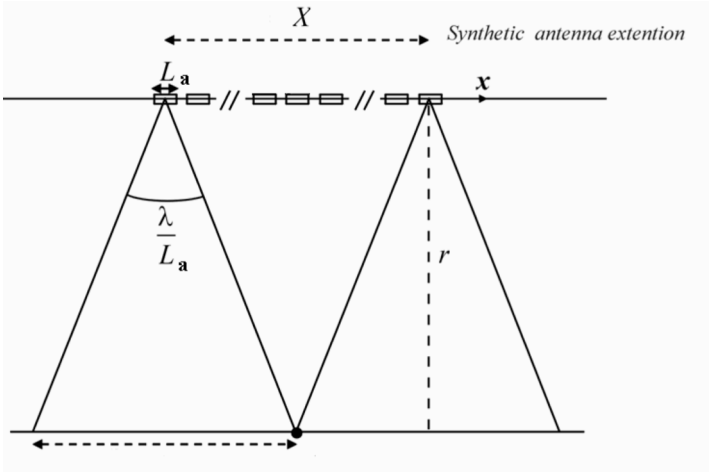


Figure 1.3: Representation of the synthetic aperture formation: the radar sensor synthesizes a larger antenna via an alignment of short aperture ones.

(called chirp, see Figure 1.2)

$$p(t) = \cos\left(2\pi f_0 t + \frac{\alpha}{2}t^2\right) rect\left(\frac{t}{\tau}\right) \tag{4}$$

where f_0 is the carrier frequency, α the so-called chirp rate and $rect\left(\frac{t}{\tau}\right)$ stands for a rectangular pulse of width τ ; it can be easily shown that, for large values of $\alpha\tau^2$, we will have

$$\alpha\tau = 2\pi\,\Delta f \tag{5}$$

that, if substituted in equation (2), leads to the following slant range resolution expression

$$\Delta r_{SR} = \frac{c}{2\,\Delta f} \simeq \frac{\pi c}{2\alpha\tau} \tag{6}$$

Therefore, the use of pulses with larger durations ($\tau = 10^{-6}\, sec$), coupled with very large chirp rates ($\alpha = 10^{14}\, rad \cdot sec^{-2}$), allows achieving slant resolutions of some meters (for example, by referring to the ERS case, we will have $\Delta r = 9.6\, m$).

In the direction orthogonal to the radar beam (azimuth), optical sensors and real aperture radars (RARs) obtain their resolution through the physical dimensions of their aperture. They correspond to the antenna azimuth footprint X, related to the antenna beam width $\lambda\, L_a$ by means of this relation

$$\Delta x_{RAR} = X = r\frac{\lambda}{L_a} \tag{7}$$

where L_a is the (actual) antenna dimension along the azimuth direction. To give an idea of the achievable azimuth resolutions, let us apply equation (7) to the ERS sensor parameters collected in Table 1.1: the achievable azimuth resolution will be of the order of some kilometers that is, obviously, not acceptable for most applications. One way to obtain better azimuth resolutions is to reduce the operational wavelength and/or to increase the antenna dimension. Besides, the wavelength value is constrained by the system characteristics and the increase of the antenna dimension is not an easy task to be accomplished, unless we implement the synthetic antenna (or aperture): "a very large antenna is synthesized by moving along a reference path a real one of limited dimension" (see Figure 1.3). The synthesis is carried out by coherently combining the back-scattered echoes received and recorded along the flight path. The improvement on the achievable azimuth resolution can be explained by observing that the extension of the synthesized array of antennas is equal to the antenna azimuth footprint X, which will correspond, accounting of the round-trip path of the signal, to the equivalent antenna beam width $\lambda\, 2X$, so that the relevant azimuth resolution can be expressed as

$$\Delta x_{SAR} = r\frac{\lambda}{2X} = \frac{L_a}{2} \tag{8}$$

As a result, the azimuth resolution is independent of the target-to-sensor distance, and is a function of the (actual) antenna dimension along the azimuth direction (that is of the order of some meters), only.

1.2.1 SAR image formation

This sub-section will address the key aspects of the SAR imaging procedure that leads to generate a complex SAR image of the illuminated area. In particular, we will refer to the more generalized "squinted SAR acquisition geometry", depicted in Figure 1.4, characterized by the presence of an offset pointing angle φ (referred to as squint angle) from the radar antenna with respect to the direction perpendicular to the flight path. Let us consider a cylindrical reference system with the axis coincident to the flight direction and denote with:

- (x, r) the azimuth and (slant) range coordinates of the generic illuminated scattering point P, respectively;
- $\theta(x, r)$ the soil surface equation in cylindrical coordinates (that is to say look angle);
- R the target-to-antenna distance relevant to the generic antenna position.

Moreover, let us consider a radar system working with a single antenna in a synchronized transmit/receive mode and, for sake of simplicity, suppose it transmits and receives the same pulse at the same position (*stop and go approximation*).

First of all, to investigate the radar signal characteristics, let us refer to the recorded echo (raw data) corresponding to a generic target of SAR coordinates (x, r) . Accordingly, we think of the SAR system as transmitting,

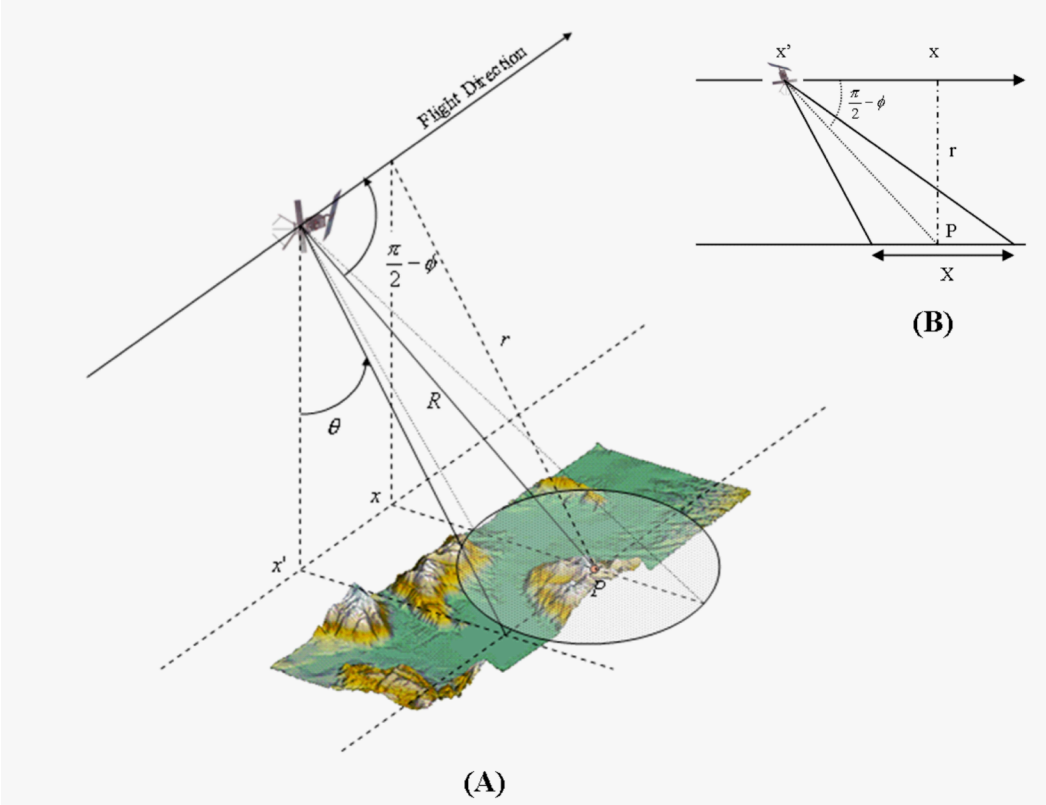


Figure 1.4: Squinted SAR reference geometry. A) Three-Dimensional representation, B) the perspective view on the equi-range plane containing the flight path and the generic target P.

at the generic instant $t_n - \tau$, a chirp pulse (see equation (4)), which can be easily treated by introducing the following complex representation

$$s(t) = exp\left[j\left(2\pi f_0(t - t_n) + \frac{\alpha}{2}(t - t_n)^2\right)\right] rect\left(\frac{t - t_n}{\tau}\right) \quad (9)$$

The relevant backscattered and onboard received signal, obtained by taking account of the transmitted/received signal time delay, will be hence expressed as follows

$$s_{received}(t) = exp\left\{j\left[2\pi f_0\left(t - t_n - \frac{2R}{c}\right) + \frac{\alpha}{2}\left(t - t_n - \frac{2R}{c}\right)^2\right]\right\} \cdot rect\left(\frac{t - t_n - \frac{2R}{c}}{\tau}\right) \quad (10)$$

and, neglecting the fast-varying $exp(j2\pi f_0 t)$ term (because it is cancelled by the heterodyne receiver) and introducing the coordinate transformation $t' = t - t_n = 2r'/c$ to manage with spatial coordinates, only, we will have

$$s_{received}(x' - x, r' - r, r) = \gamma(x,r) exp\left[j\left(-2\pi f_0 \frac{2R}{c} + \frac{\alpha}{2}\left(\frac{2r' - 2R}{c}\right)^2\right)\right] \cdot rect\left(\frac{\frac{2r' - 2R}{c}}{\tau}\right) \cdot w^2\left(\frac{x' - x - r\tan\phi}{\frac{v_s}{X}}\right) \quad (11)$$

where:

- $x' = v_s t_n$ represents the azimuth coordinate of the antenna phase center;
- $\gamma(x,r)$ is the reflectivity function of the target P, which is proportional to the ratio between the backscattered and the incident field;
- $w^2(\cdot)$ is the antenna illumination function, related to the azimuth antenna footprint over the ground, properly squared to account that the same antenna operates both in receiving and transmitting mode;

- X is the antenna azimuth footprint, whose general expression, taking account of the squint angle (see Figure 1.4B) is now $X = \lambda r L_a cos^2\varphi$.
- $R = r + \Delta R$

Given that a comprehensive description of the overall SAR image processing step (usually referred to as focusing procedure) is outside the scope of this work, we will present the final result of the received signal and refer the reader back to [14] for a more detailed one, in particular we have that:

$$S_{received}(x' - x, r' - r, r) = \tilde{\gamma}(x, r) \cdot g(x' - x, r' - r, r) \tag{12}$$

where:

- $\tilde{\gamma}(x, r) = \gamma(x, r) \cdot exp\left[-j\frac{4\pi}{\lambda}r\right]$ is the product of the scene reflectivity function and the round-trip phase term (representing the core for the InSAR applications);
- $g(x' - x, r' - r, r)$ is the SAR raw-data impulse response.

Finally, the focusing procedure correspond in an estimation of the reflectivity function of the illuminated ground scene, in particular:

$$\hat{\gamma}(x', r') = \iint \gamma(x, r) exp\left(-j\frac{4\pi}{\lambda}r\right) sinc\left[\frac{\pi}{\Delta r}(r' - r)\right] \cdot \\ \cdot sinc\left[\frac{\pi}{\Delta x}(x' - x)\right] exp\left[j\xi_{dopp}(x' - x)\right] dx dr \tag{13}$$

where:

- the $sinc(\cdot)$ function, whose role will be essential to estimate the achievable spatial resolution of a SAR system, is the so-defined one: $sinc(\cdot) = \frac{\sin(x)}{x}$ and is characterized by a 3dB-main lobe aperture;
- ξ_{dopp} is the central angular frequency value, which, properly converted in Hz, leads to the expression of the so-defined Doppler centroid frequency;
- $\Delta x = \frac{L_a}{2}$ and $\Delta x = \frac{c}{2\Delta f}$ are the achievable azimuth and range resolution expression, respectively.

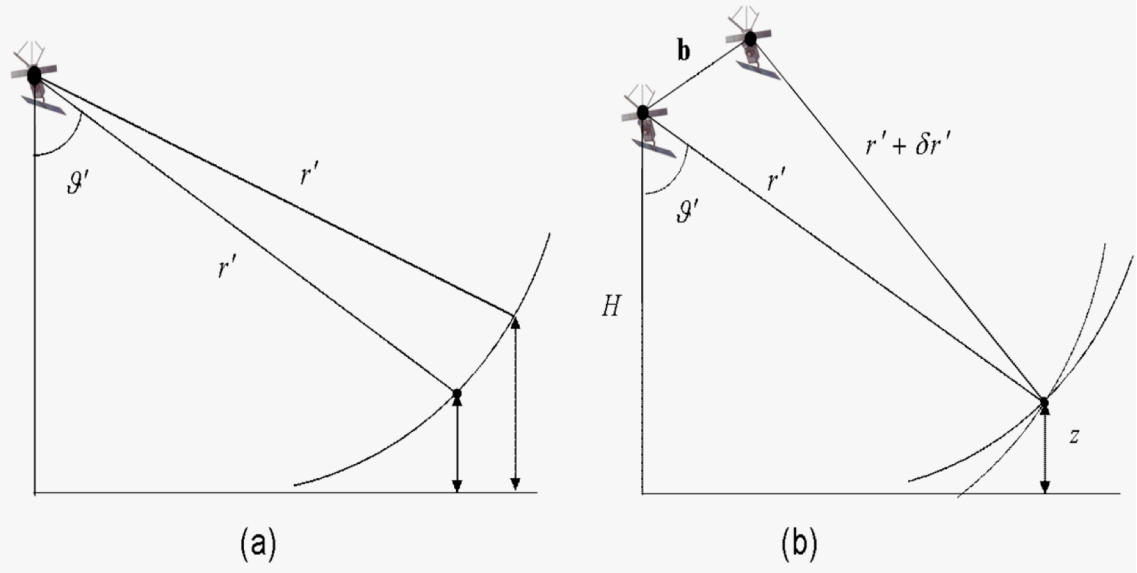


Figure 1.5: Geometry in the plane orthogonal to the flight direction with a single sensor (a) and in the stereometric (b) case.

The focused SAR image can be expressed and managed as a complex-valued matrix, and each resolution cell corresponds to a pixel on the SAR image. In addition each single SAR image can be also effectively analyzed via the inspection of the relevant amplitude and phase component, respectively. The amplitude is related to the backscattering coefficient of the ground, while the phase contains information about the travel distance of the emitted signal.

1.3 Interferometry

Interferometric technique [15] exploits the phase difference of two complex SAR images, acquired from different orbit positions and/or at different times, to estimate the position on the ground of the illuminated targets. In principle, different antenna distributions define different interferometer configurations. In particular, the InSAR configurations involve two antennas observing the investigated scene:

- at the same time and from different positions, spaced in the across- track direction (across-track interferometry);
- at different times and from the same position (along-track interferometry);
- at different times and from different positions (repeat-pass across- track and along-track interferometry).

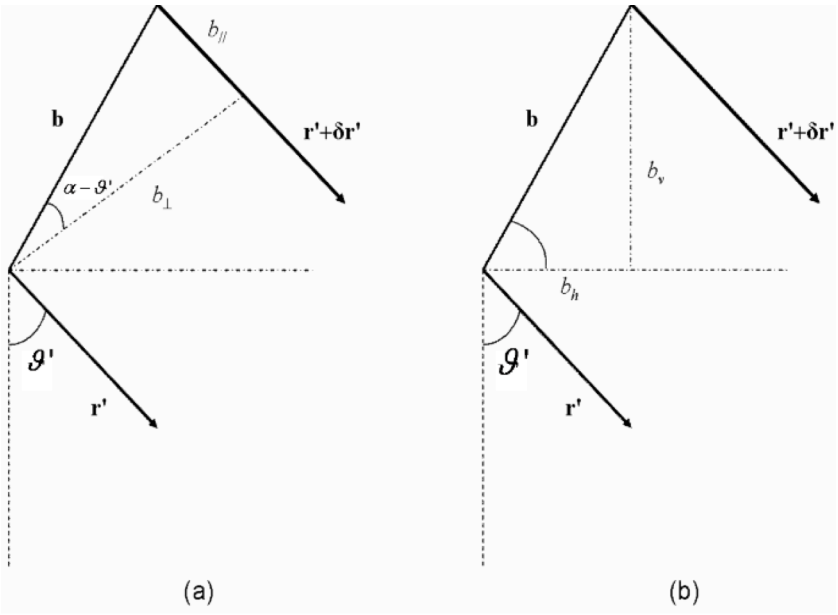


Figure 1.6: Interferometric baseline geometry. The baseline vector can be effectively decomposed in the perpendicular/parallel components (a) and in the horizontal/vertical components (b), respectively.

We will focus, in particular, on the across track repeat-pass interferometry, suitable for space-borne applications, whose general configuration in the plane orthogonal to the azimuth direction is the same used within the stereometric analysis (see Figure 1.5 b). In this plane, the two SAR antennas are separated by a baseline vector **b** which can be decomposed into either parallel/perpendicular ($b_{//}, b_{\perp}$) components or horizontal/vertical ones (b_h, b_v), respectively. It is easy to calculate them through the inspection of Figure 1.6, thus obtaining

$$\begin{cases} b_{\perp} = bcos(\theta' - \alpha) \\ b_{//} = -bsin(\theta' - \alpha) \\ b_h = bcos(\alpha) \\ b_v = bsin(\alpha) \end{cases} \tag{14}$$

where θ' is the side-looking angle and α is the elevation angle. The two focused SAR images may be represented (see equation 13) via the following expressions (being $\gamma_1(\cdot)$ and $\gamma_2(\cdot)$ the two distinctive reflectivity function of the illuminated area at the two different acquisition times)

$$\begin{aligned} \hat{\gamma}_1(x', r') = \iint \gamma_1(x, r) exp\left(-j\frac{4\pi}{\lambda}r\right) sinc\left[\frac{\pi}{\Delta r}(r' - r)\right] \cdot \\ \cdot sinc\left[\frac{\pi}{\Delta x}(x' - x)\right] exp[j\xi_{d1}(x' - x)] dx dr \end{aligned} \tag{15}$$

$$\begin{aligned} \hat{\gamma}_2(x', r') = \iint \gamma_2(x, r) exp\left(-j\frac{4\pi}{\lambda}r\right) sinc\left[\frac{\pi}{\Delta r}(r' - r)\right] \cdot \\ \cdot sinc\left[\frac{\pi}{\Delta x}(x' - x)\right] exp[j\xi_{d1}(x' - x)] dx dr \end{aligned}$$

For the time, let us assume the $sinc(\cdot)$ functions could be represented as Dirac ones (in other words, we will first refer to the non-realistic case characterized by an infinite bandwidth, corresponding to a zero-valued spatial resolution). In this way, the measured reflectivity returns associated to the two SAR images (regardless to the pixel of SAR coordinates (x', r') can be expressed as follows:

$$\hat{\gamma}_1(x', r') = \gamma_1(x', r') exp\left(-j\frac{4\pi}{\lambda}r'\right)$$

$$\hat{\gamma}_2(x', r') = \gamma_2(x', r' - \delta r') exp\left(-j\frac{4\pi}{\lambda}r'\right) \tag{16}$$

The misalignment of the second image (usually referred to as *slave* image) with respect to the first one (*master* image), is clearly due to the path difference between the two radar signals. For this reason, it is needed to properly register the images each other, thus working with the same reference geometry

$$\hat{\gamma}_2(x', r') \rightarrow \hat{\gamma}_2(x', r' + \delta r') = \gamma_2(x', r') exp\left(-j\frac{4\pi}{\lambda}(r' + \delta r')\right) \tag{17}$$

From the equations (15) and (16), we generate the complex interferogram, so defined:

$$\hat{\gamma}_1(x', r')\hat{\gamma}_2^*(x', r') = |\hat{\gamma}_1(x', r')| \cdot |\hat{\gamma}_2(x', r')| e^{j\frac{4\pi}{\lambda}\delta r' + j\angle\gamma_1(x', r') - j\angle\gamma_2(x', r')} \tag{18}$$

where the $*$ operator denotes the complex conjugate operation, while $\angle(\cdot)$ refers to the full phase (i.e., the phase value is not restricted to the $]-\pi, \pi]$ interval) extraction operation. Assuming that the scattering mechanism on the ground has not changed between the two passages of the sensor over the

illuminated area (this requirement could be directly accomplished if a single- pass configuration was considered), the measured phase difference (being $\angle\gamma_1(x',r') = \angle\gamma_2(x',r')$) will depend on the imaging geometry, only

$$C(x',r') = |\gamma_1^2(x',r')|^2 exp\left(-j\frac{4\pi}{\lambda}\delta r'\right) \tag{19}$$

whose full phase term is

$$\Delta\Phi = \frac{4\pi}{\lambda}\delta r' \tag{20}$$

Taking into account the equations (14) and the acquisition geometry depicted in Figure 1.5, we get

$$r' + \delta r' = \sqrt{b^2 + r'^2 - 2br'sin(\theta' - \alpha)} \approx r' - bsinn(\theta' - \alpha) \tag{21}$$

where the parallel ray approximation introduced by Zebker and Goldstein [3] has been applied. Hence, the interferometric phase can be written as follows

$$\Delta\Phi = \frac{4\pi}{\lambda}\delta r' \cong \frac{4\pi}{\lambda}bsin(\theta' - \alpha) = \frac{4\pi}{\lambda}b_{//} \tag{22}$$

To give an example, in Figure 1.7 a tandem interferogram (whose time delay is equal to one day) is shown. Since the phase difference can be measured only in the interval $[-\pi, \pi[$, it clearly exhibits several fringes, corresponding to the $2\pi-$phase jumps.

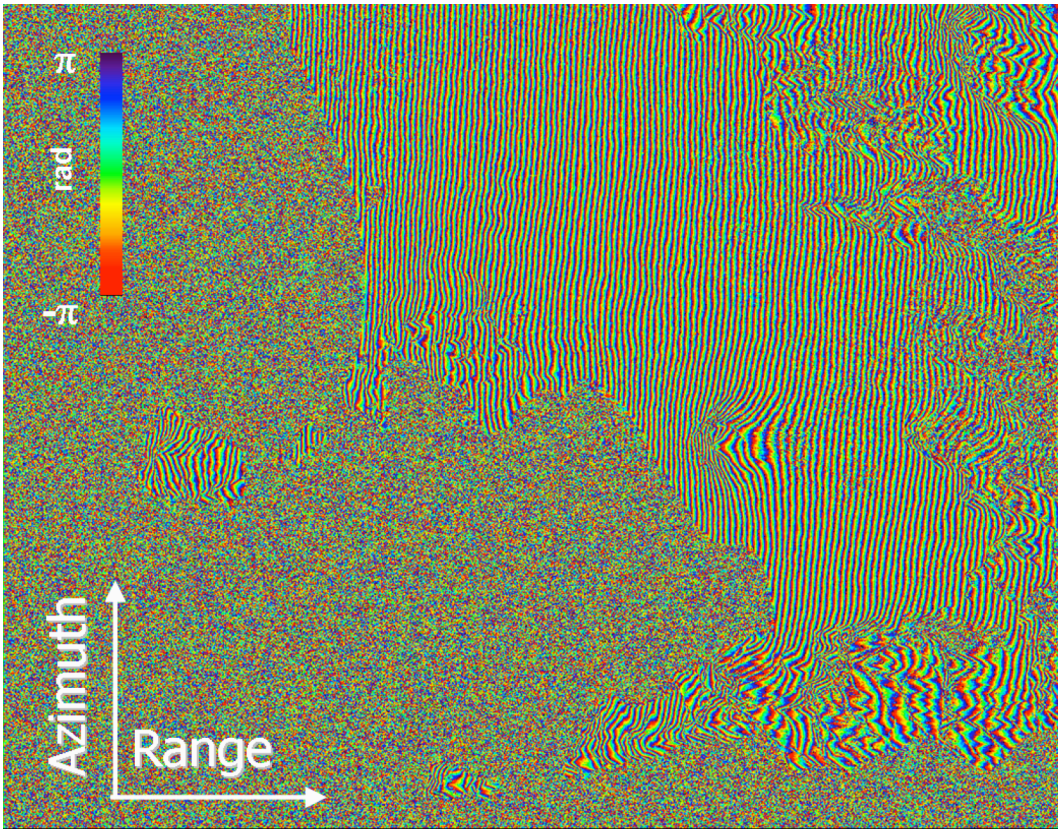


Figure 1.7: The tandem interferogram computed from the ascending ERS data pair acquired on August 1, 1995 and August 2, 1995. The flat-Earth term is responsible for the fast-varying fringe pattern along the range direction, modulated by the topographic components, itself

As a matter of fact the full phase term $\Delta\Phi$ is composed by two fundamental terms, whose mathematical description is extensively proposed in [14], however the complete expression of the interferometric phase is the follow:

$$\Delta\Phi(x', r') \approx \Delta\Phi^{flat}(x', r') - \frac{4\pi}{\lambda}\frac{b_{\perp}}{r' \sin\theta'} z \tag{23}$$

$$\Delta\Phi^{flat}(x', r') = -\frac{4\pi b_{\perp}}{\lambda r' \tan\theta'_0}(r' - r'_0) \tag{24}$$

The first term of the equation (23) accounts for the *flat-Earth* contribution and the second one for the topographic profile, respectively. To give an example, in Figure 1.7 a tandem interferogram (whose time delay is equal to one day) is shown. Since the phase difference can be measured only in the interval $[-\pi, \pi[$, it clearly exhibits several fringes, corresponding to the 2π-phase jumps. The *flat-Earth* contribution, in particular, is responsible for the rapid phase change, which increases in the range direction (clearly visible in Figure 1.7).

1.4 Deformation derivation from Interferometry (DInSAR)

If two acquisitions are made from the same position but at different times any changes in the target's position occurred during this time can be determined. This leads to the conclusion that the spatial perpendicular baseline must be as small as possible. Then the target will be viewed with the same geometry two times and if the distance radar-terrain-radar leaves identical in both acquisitions then the phases of returned scatterer will be the same which means the difference between them will be zero. If the terrain has shifted (up, down, lateral) between these two acquisitions changing the distance radar-terrain-radar then the phase in the second image will be also shifted. This shift indicates the terrain movements. The amount of movement corresponding to the phase cycle is determined from the Equation (6) as:

$$\Delta r = \frac{\lambda}{2} \tag{25}$$

It implies that half wavelength (28mm for ERS) change in the range due to the deformation creates one fringe in the interferogram.

Therefore, we introduce a deformation component on the measured phase difference. By referring to the geometry depicted in Figure 1.8, we will be able to compute the phase difference

$$\Delta\Phi^{def} = \frac{4\pi}{\lambda}(r_2 - r_1) = \frac{4\pi}{\lambda} d_{LOS} = \frac{4\pi}{\lambda} d \sin(\theta' - \alpha_D) \tag{26}$$

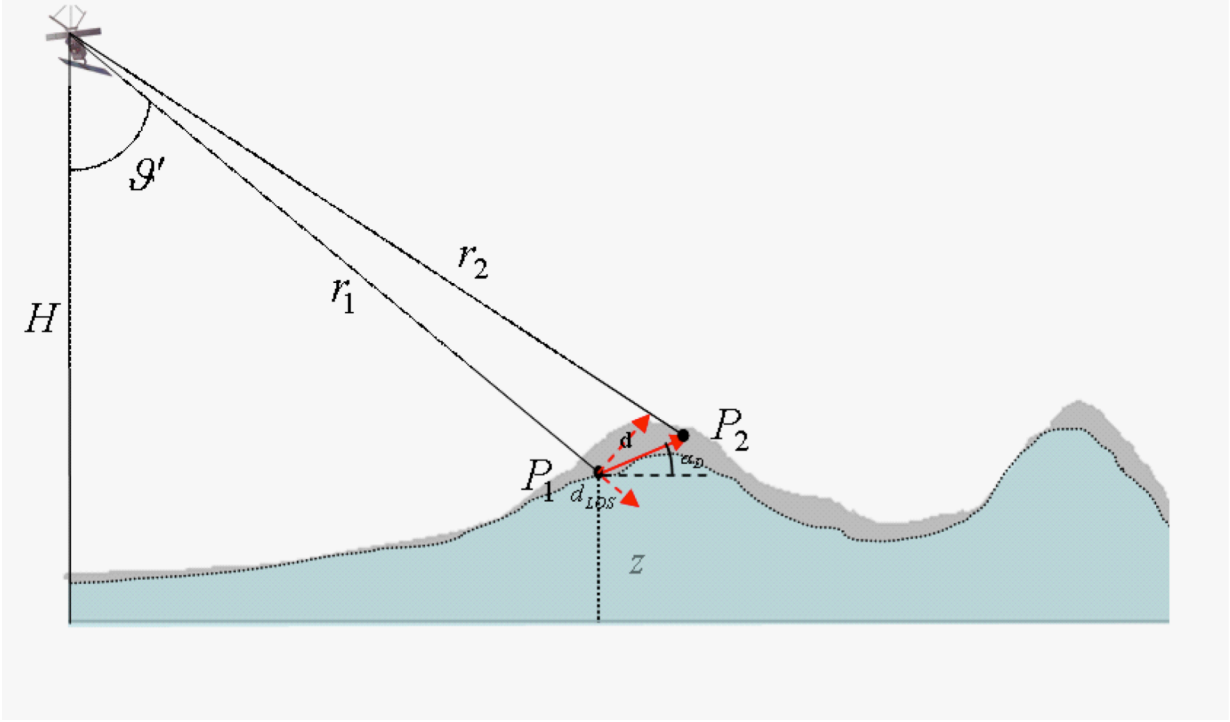


Figure 1.8: The simplified interferometric geometry where the master and slave orbital positions are considered to be identical, thus allowing immediately individuating the deformation component. P_1 represents the position of the given target on the ground viewed prior the deformation has occurred, while P_2 refers to the position of the same target after the deformation episode (the magnitude of the deformation has been intentionally exaggerated to better show the geometry of the system).

In this way, the equation (26) will allow measuring the displacement component that is parallel to the look angle direction (usually referred to as Line Of Sight (LOS) displacement). In this ideal configuration, the D-InSAR technique gets an unambiguously measurement of the LOS displacement

of the order of fractions of wavelength: indeed, a differential phase change of 2π is now associated to a LOS displacement of $\lambda/2$.

When a no zero perpendicular baseline between the orbits are considered, additional topographic fringes occurs. In this case, by referring to Figure 1.9, the phase difference will be :

$$\Delta\Phi^{def} = \frac{4\pi}{\lambda}(r_2 - r_1) = \frac{4\pi}{\lambda}(\tilde{r} - r_1) = \frac{4\pi}{\lambda}(r_2 - \tilde{r}) =$$

$$= \Delta\Phi^{flat} + \Delta\Phi^{tpg} + \Delta\Phi^{def} \qquad (27)$$

The topographic phase term is not of interest with regard to the surface displacement mapping.

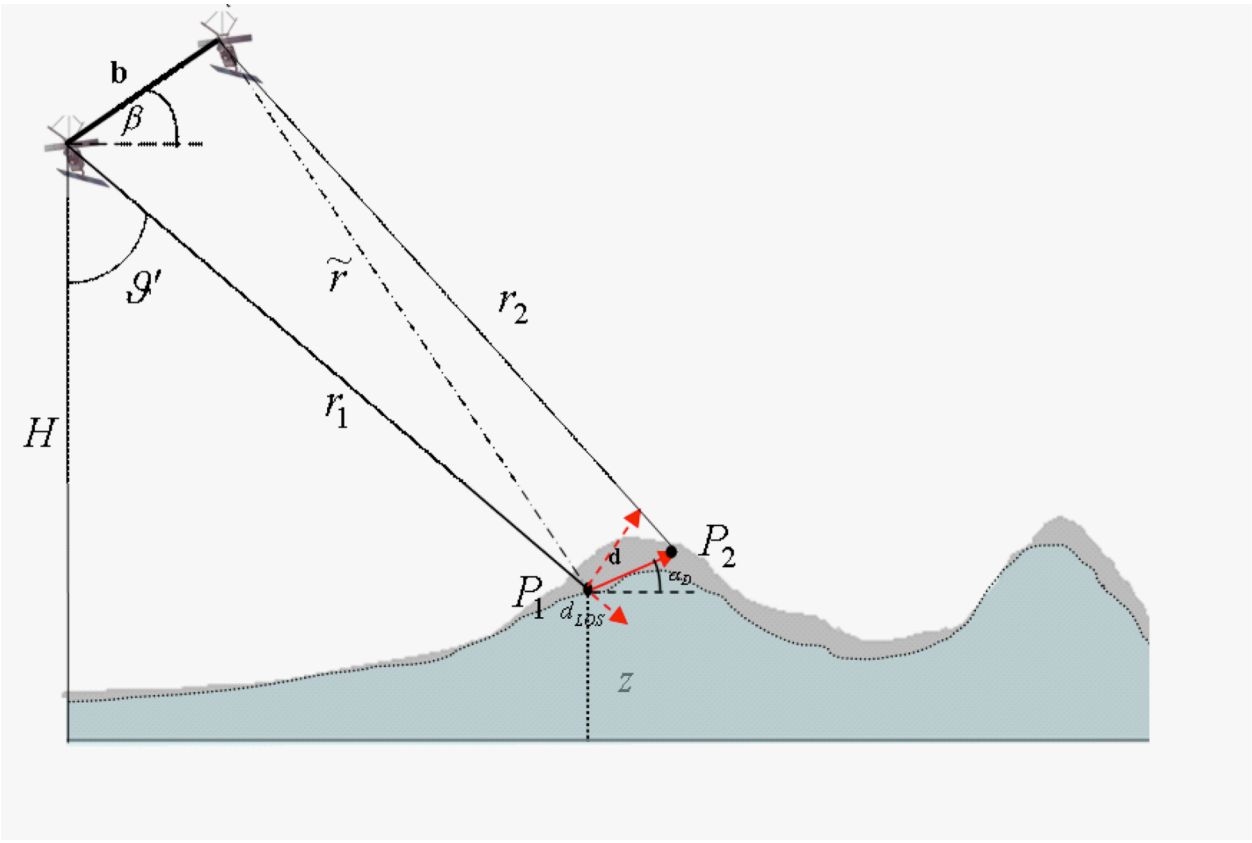


Figure 1.9: The interferometric geometry where the slant range distances responsible for the topographic and the

For this reason, in order to limit its influence, equation (67) would suggest the use of small perpendicular baselines, only. The discrimination between the topographic and the displacement phase terms, respectively, must be carried out. There are several approaches to measure the topographic components, in particular in SBAS processing chain the topographic phase is calculated through an external, conventional DEM (Digital Elevation Model). To do this, the DEM has to be transformed in the azimuth- range geometry, scaled with respect to the baseline of the interferogram and the resulting topographic pattern have to be finally converted from length to phase units.

Thus, if topography is known we can determine the deformation. The method in which topography contributed signal is subtracted from the interferometric phase is called *differential interferometry*.

InSAR measurable ground changes can be due to the earthquakes, volcanoes, land subsidence, tectonic, or ice, glacier movements, or anthropogenic influences such as mining, explosions. [15].

The feasibility of differential SAR interferometry for detecting ground displacement depends on the characteristics of terrain, deformation, its magnitude and the spatial extend over which it occurs, the time interval between the image acquisitions with respect to the deformation size. Also it depends on the characteristics of the system, how long the signal's wavelength, how large the swath, pixel size.

The pixel on radar image contains hundred elementary targets. The phase sum of these hundreds targets represents the pixel's phase. The main condition of interferometry is that the phase contribution of each elementary targets remain the same between the two phase images. However, complete stability is not possible and the question is what amount of their changes can measure SAR interferometry. If the relative positions of elementary targets within the pixel change by amounts greater than the radar wavelength, then coherence is lost and the measurement is impossible. On the contrary if the change is coherent common to the several adjacent pixels, then the phase comparison is possible. The maximum detectable deformation gradient is one fringe per pixel, or the dimensionless ratio of the pixel size to the wavelength.

However, it is possible to estimate the relative range change for two points within an interferogram, by integrating the number of fringes between them. This process of estimating the integrated phase is known as phase-unwrapping.

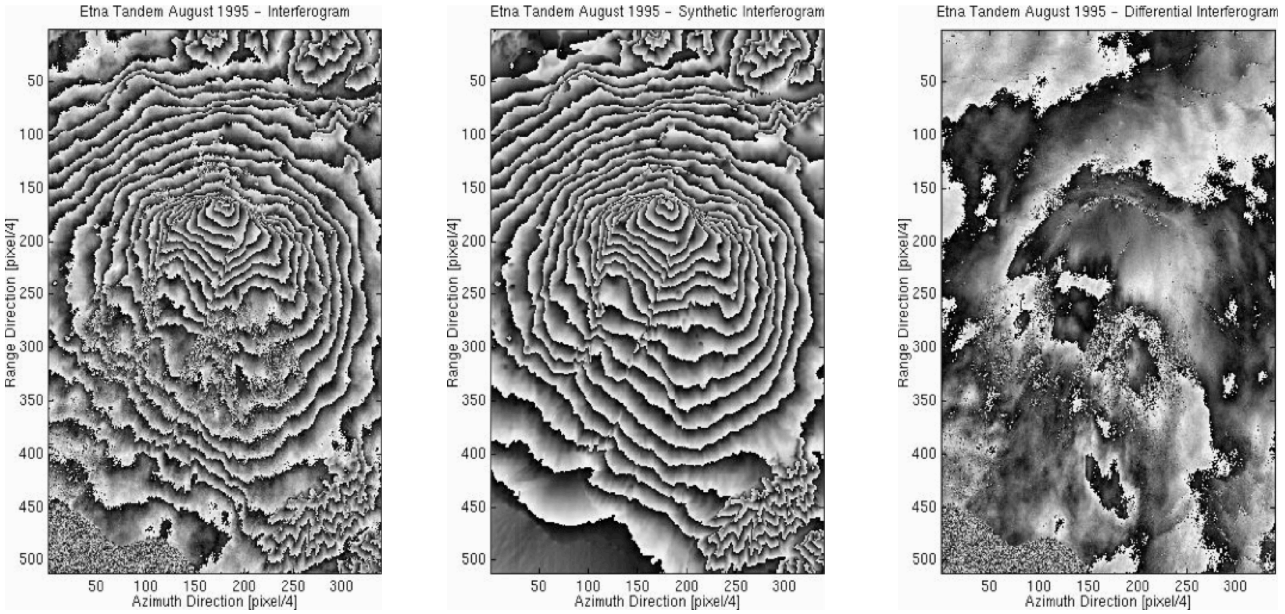


Figure 1.8: On the left ids shown a picture of a real Interferogram; in the center a reconstruction of the same interferogram using the range information (synthetic Interferogram); On the right the result of a the difference between the real interferogram with the synthetic one (Differential Interferogram)

1.5 Time series InSAR

A big limitation of InSAR, particularly in the case of small strain, due to processes such as interseismic plate motion and glacio-isostatic adjustment, is that the non-deformation signal can swamp the deformation signal. One approach for extracting small displacements is summing or "stacking" of many conventionally formed interferograms [16, 17, 18]. The deformation signal reinforces, whereas

other signals typically do not. However this approach is only appropriate when the deformation is purely steady-state, with no seasonal deformation, and even then is not optimal, as the non-deformation signals are reduced only by averaging rather than by explicit estimation.

Algorithms for time series analysis of SAR data have been developed to better address these two limitations of conventional InSAR. The first limitation is tackled by using phase behaviour in time to select pixels where decorrelation noise is minimised. The second limitation is addressed by estimating the non-deformation signal by a combination of modelling and filtering of the time series. The time series algorithms fall into two broad categories, the first being Persistent Scatterer InSAR, which targets pixels with consistent scattering properties in time and viewing geometry, and the second being the more general Small Baseline approach.

In this work we refer to the Small Baseline approach and in particular to Small BAseline Subset technique, an interferometric method for the displacement time series genration, completely developed at IREA CNR of Naples.

1.5.1 Small BAseline Subset (SBAS) technique

For resolution elements containing no dominant scatterer, phase variation due to decorrelation is often large enough to obscure the underlying signal. However, by forming interferograms only between images separated by a short time interval and with a small difference in look and squint angle, decorrelation is minimised, and for some resolution elements can be small enough that the underlying signal is still detectable. Decorrelation is further reduced by spectral filtering in range (Gatelli et al., 1994) and discarding of the non-overlapping Doppler frequencies in azimuth. Pixels for which the filtered phase decorrelates little over short time intervals are the targets of small baseline methods.

Interferograms are formed between SAR images that are likely to result in low decorrelation noise, in other words, those that minimise the difference in time, look angle and squint angle (see Figure 1.9). Obviously it is not possible to minimise each of these simultaneously, so assumptions have to be made about the relative importance, based on the scattering characteristics of the area of interest. In many small baseline algorithms, the interferograms are then multilooked to further decrease decorrelation noise [19, 20] .

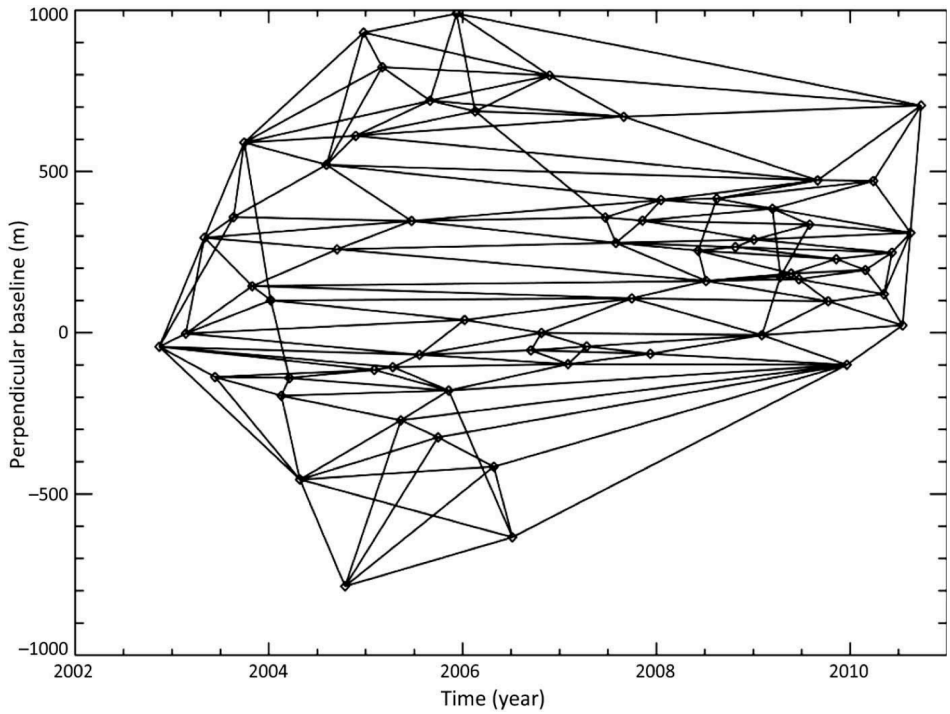


Figure 1.9: An example of distribution of the a SAR dataset in the temporal/perpendicular baseline plane according to the Small Baseline approach.

Pixels are selected based on their estimated spatial coherence n each of the interferograms, using either standard coherence estimation [21]. The phase is then unwrapped using the additional dimension of time in 3-D approaches [22]. At this point the phase can be inverted to give the phase at each acquisition time with respect to a single image, using singular value decomposition [19]. Separation of

deformation and atmospheric signals can be achieved by filtering the resulting time series in time and space.

The small baseline method can achieve accuracies on the order of ~1 mm/year [23], although this again depends on the number of images, the time over which the images are acquired and the distance from the reference point or area.

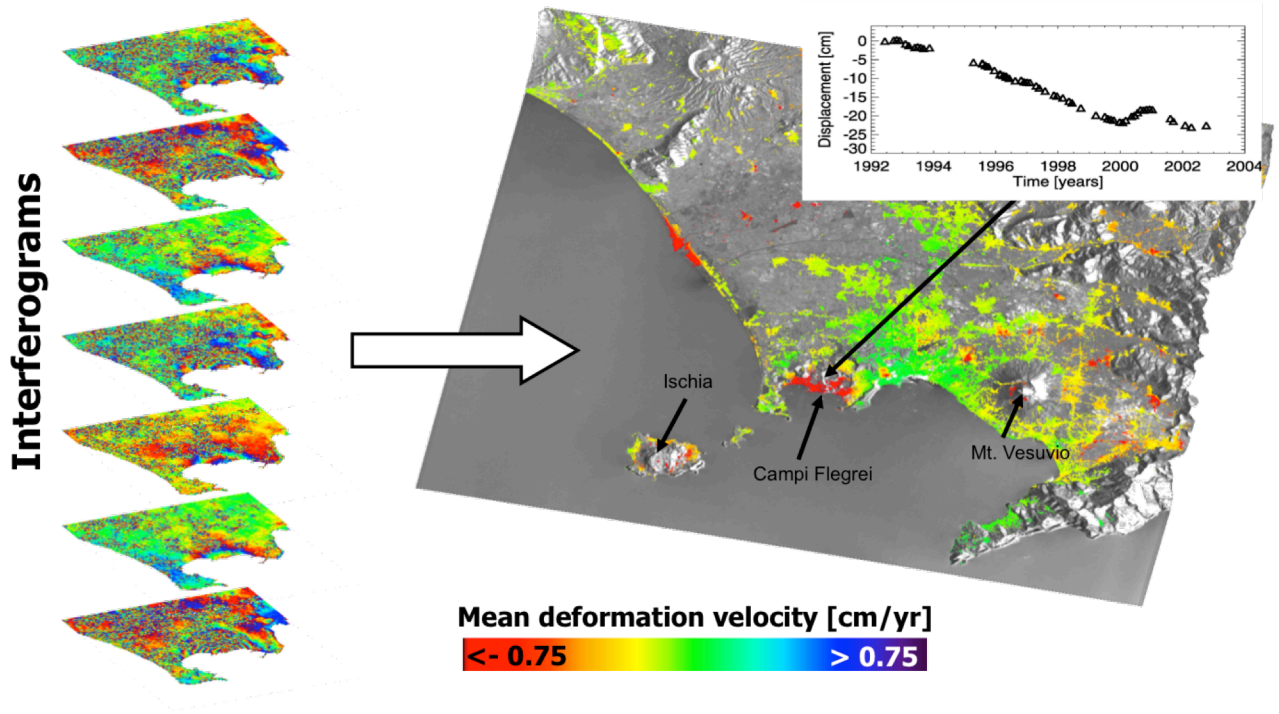


Figure 1.10: conceptual representation of the operating principle of SBAS technique. On the left a differential interferogram stack from which is possible to obtain the temporal evolution of the deformation. On the right the mean deformation velocity map over Napoli Bay Area obtained with SBAS technique by using ERS and ENVISAT sensor. A plot of time series of a pixel affected by high deformation phenomena is shown in the upper right of the Figure .

References

[1] C. A. Wiley: “Pulsed Doppler radar methods and apparatus”, U.S. Patent 3,196,436, filed in 1954;

[2] S. H. Pravdo, et al: "Seasat Synthetic Aperture Radar Data User's Manual," JPL publication 82-90, March 1, 1983;

[3] H. A. Zebker and R. M. Goldstein: “Topographic mapping from synthetic aperture radar observations”, J. Geophys. Res., 91, 4993, 1986;

[4] K. Gabriel, R. M. Goldstein and H. A. Zebker: “Mapping small elevation changes over large areas: Differential interferometry”, J. Geophys. Res., 94, 1989;

[5] J. W. Moore, "OSTA-1: The Space Shuttle's first scientific payload," Proceedings of 33rd IAF/IAC Congress, Paris, France, Sept. 27-Oct. 2, 1982;

[6] Pettengill, H. Gordon; Ford, G. Peter; Johnson, T. K. William; R. Raney, Keith, Soderblom, A. Laurence: “Radar performance and data products”, Science, vol. 252, April 12, 1991, p. 260-265;

[7] Special Issue on SIR-C/X-SAR, IEEE Transactions on Geoscience and Remote Sensing, Vol. 33, No. 4, July 1995;

[8] European Space Agency: “ERS SAR RAW: ERS-1/2 synthetic aperture radar (SAR) annotated raw product data”, http://earth1.esrin.esa.it:81/, 1998. Earthnet online;

[9] R. L. Jordan, E. R. Caro, Y. Kim, M. Kobrik, Y. Shen, F. V. Stuhr and M. U. Werner: “Shuttle radar topography mapper (SRTM) Microwave Sensing and Synthetic Aperture Radar”, Proc. SPIE, pp 412-422;

[10] H. Yoshihisa and O. Makoto: “Analysis of JERS-1 SAR imagery”, in International Geoscience and Remote Sensing Symposium, Tokyo, Japan, 18-21 August 1993, pages 1191-1193, 1993;

[11] http://envisat.esa.int/instruments/asar

[12] M. Shimada, A. Rosenqvist, M. Watanabe, T. Tadono: “Polarimetric and interferometric potential of the PALSAR/ALOS”, in Proc. POLinSAR, 2005;

[13] F. De Zan and A. M. Monti Guarnieri, "TOPSAR: Terrain Observation by Progressive Scans," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 9, pp. 2352-2360, Sept. 2006., doi: 10.1109/TGRS.2006.873853

[14] G. Franceschetti and R. Lanari: “Synthetic Aperture Radar Processing”, CRC PRESS, New York, 1999;

[15] D. Massonnet and K. Feigl. Radar interferometry and its application to changes in the earth’s surface. *Reviews of Geophysics*, 36(4):441–500, No- vember 1998.

[16] Simons, M., Rosen, P.A., 2007. Interferometric synthetic aperture radar geodesy. In: Schubert, G. (Ed.), Treatise on Geophysics, Vol. 3. Elsevier Press, pp. 391–446.

[17] Schmidt, D.A., Bürgmann, R., 2003. Time-dependent land uplift and subsidence in the Santa Clara valley, California, from a large interferometric synthetic aperture

[18] Zebker, H.A., Rosen, P.A., Goldstein, R.M., Gabriel, A., Werner, C.L., 1994. On the deri- vation of coseismic displacement-fields using differential radar interferometry — the Landers earthquake Oct Journal of Geophysical Research 99 (B10), 19,617–19,634.

[19] P. Berardino, G. Fornaro, R. Lanari and E. Sansosti: “A new Algorithm for Surface Deformation Monitoring based on Small Baseline Differential SAR Interferograms”, IEEE Trans. Geosci. Remote Sens., 40, 11, 2002;

[20] Schmidt, D.A., Bürgmann, R., 2003. Time-dependent land uplift and subsidence in the Santa Clara valley, California, from a large interferometric synthetic aperture

[21] Li, F.K., Goldstein, R.M., 1990. Studies of multibaseline spaceborne interferometric syn- thetic aperture radars Jan IEEE Transactions on Geoscience and Remote Sensing 28 (1), 88–97.

[22] Pepe, A., Lanari, R., 2006. On the extension of the minimum cost flow algorithm for phase unwrapping of multitemporal differential SAR interferograms. IEEE Transac- tions on Geoscience and Remote Sensing 44 (9), 2374–2383.

[23] Lanari, R., Casu, F., Manzo, M., Zeni, G., Berardino, P., Manunta, M., Pepe, A., 2007. An overview of the small baseline subset algorithm: a DInSAR technique for surface deformation analysis. Pure and Applied Geophysics 164 (4), 637–661.

Chapter 2

SBAS-DInSAR Parallel Processing for Deformation Time Series Computation

The aim of this chapter is to describe a novel parallel computing solution for the processing chain implementing the Small BAseline Subset (SBAS) Differential SAR Interferometry (DInSAR) technique. The developed parallel solution (P-SBAS) is based on a dual-level parallelization approach and encompasses combined parallelization strategies, which are fully discussed in the follow. Moreover, the main methodological aspects of the proposed approach and their implications are also addressed. Finally, an experimental analysis, aimed at quantitatively evaluating the computational efficiency of the implemented parallel prototype, with respect to appropriate metrics, has been carried out on real data and making use of an in-house cluster infrastructure; this analysis confirms the effectiveness of the proposed parallel computing solution. In the current scenario, characterized by huge SAR archives relevant to the present and future SAR missions, the P-SBAS processing chain can play a key role to effectively exploit the big data volumes and High Performance Computing (HPC) platforms for the comprehension of the surface deformation dynamics of large areas of Earth.

In addition the Appendix-A gives an overview about the major HPC infrastructures (cloud, grid and cloud) that are nowadays the most important computing models for science applications. Indeed, in the following of this thesis work are described some research activities concern to the employment of these computing models for P-SBAS application.

2.1 Sequential SBAS Processing Chain

The SBAS processing chain, as for any other DInSAR algorithm dealing with data stacks, requires a number of conceptual steps that have to be sequentially performed. These, in particular, are the SAR image co-registration, the interferogram generation, the unwrapping of the computed phases, and the retrieval of the final displacement time-series, all of them possibly preceded by the SAR focusing, if raw data are concerned [27]. However, depending on the specific algorithm used at each of these steps, the final implementation of the processing chain results in a more complex sequence of operations as schematically depicted in Figure 2.1, in which the workflow of the SBAS processing chain is presented. Before providing more details on the SBAS procedure, some general notes are in order. First, the processing steps from block A to G are performed at full spatial resolution, whereas the subsequent steps work on multilooked data. Second, a common storage is assumed to be available to all the processing phases, i.e., each step gains access to the same common storage for reading inputs and

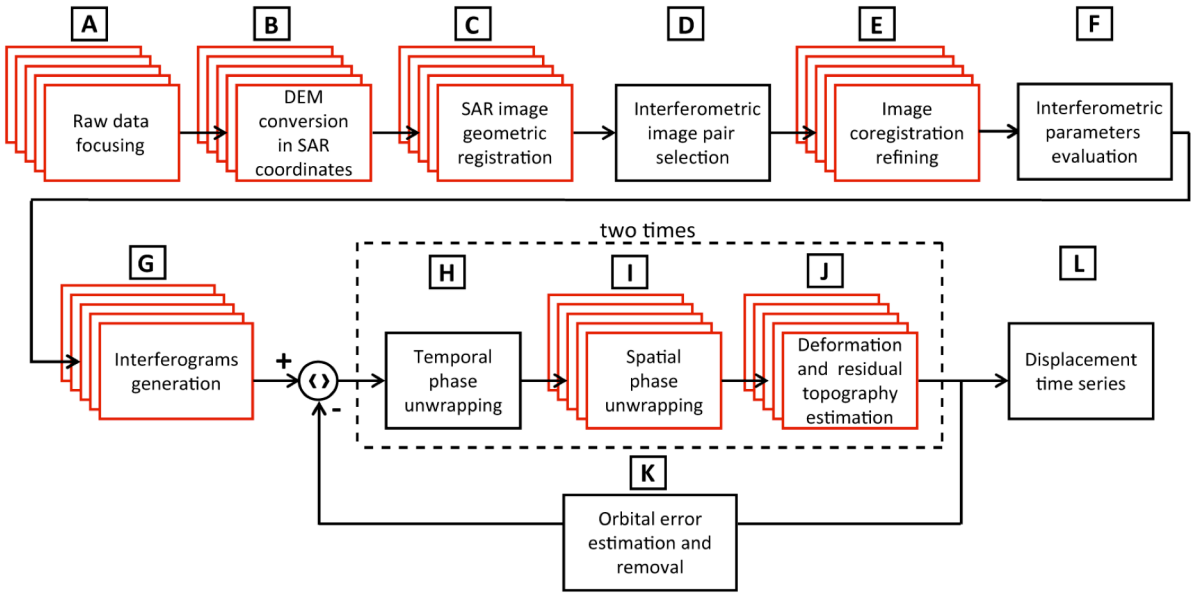


Figure 2.1: P-SBAS workflow Black/single and red/multilayered blocks represent sequential and parallel (from a process-level perspective) processing steps, respectively. This scheme reduces to the SBAS sequential workflow when only one processor is employed.

writing outputs.

Let us now concentrate on the SBAS sequential chain description. After ancillary data and acquisition parameter retrieving, the SAR raw data focusing (block A of Figure 2.1) is performed, by considering the zero Doppler geometry, according to the approach shown in [28]. This algorithm is implemented by first decomposing the range compressed raw data into overlapping blocks of a defined size (depending on the Doppler frequency variation, the range cell migration, and the synthetic antenna footprint) along the azimuth and range directions. Subsequently, each block is compressed in the Fourier domain and saved to the storage. A final step is needed for combining together all the single blocks in the space domain and obtaining a full size single look complex (SLC) image. Such an implementation permits to generate extremely accurate focused SAR images, independently from the Doppler frequency variation and replica; however, it results in a high computational burden due to the several number of performed bidimensional FFTs (fast Fourier transform) and input/output (I/O) operations. Data flow volume is also significantly high being all the operation performed in the complex domain.

After raw SAR data focusing, the area common to all the images of the stack is identified to be used in all the subsequent processing steps, as well as the master image, which imposes the reference geometry for the entire processing. On this common region, the conversion of the digital elevation model (DEM) into SAR coordinates is performed (step B of Figure 2.1) by exploiting the s called range-Doppler equations [29]. This step, which typically exploits shuttle radar topography mission (SRTM) data, first requires an oversampled DEM to be converted in a Cartesian Earth-fixed Earth-centered system. Afterward, by exploiting the orbit of the reference master image, the azimuth and range coordinates per each DEM pixel are identified and stored into two files. The latter values are provided in an irregular grid, and then a further gridding process is mandatory. Due to DEM oversampling, the dimension of the involved data matrixes can become not negligible (several gigabytes also in the canonical ERS and ENVISAT cases), particularly when wide areas are concerned, requiring high memory/disk storage for

dealing with the intermediate products. Moreover, the procedure is performed in double precision floating point to reduce the error propagation, thus further increasing the required memory/storage and the I/O access.

External DEM is needed not only for topography removal in the differential interferogram generation, but also for co-registering each image with respect to the master reference geometry. Indeed, following the approach described in [30], a geometrical co-registration step (block C of Figure 2.1) is performed after having computed, per each image, the distances along line of sight (LOS) (range) and flight track (azimuth) of the scene targets (image pixels) from an orbital reference position: the resulting archives are called range and azimuth files. These files are computed in double precision floating point via a simple geometrical solution and are later stored to be available for the subsequent co-registration step and differential interferogram generation. The co-registration step (block C of Figure 2.1), indeed, needs the mentioned range and azimuth files as input for computing the warping function as well as the slave image to be interpolated with respect to the master geometry [30]. Processing itself is not particularly heavy in terms of complexity; indeed, the warping function is straightforwardly estimated from the range and azimuth files by difference, while the slave interpolation is performed in double precision using a Knab kernel [31]. Again, the main limitation for this step is represented by the file dimension (mainly in the case of range and azimuth files), which requires intensive I/O access and available memory.

After having co-registered all the images with respect to the master one, the identification of interferometric data pairs is carried out (step D of Figure 2.1). Such identification is required for the subsequent co-registration refinement step. Indeed, for identifying possible residual rigid shifts of the geometric co-registration (due to inaccuracies on the system parameter knowledge), an iterative procedure is performed on each data pair by searching for the sub-pixel shift that maximizes the spatial coherence of an image patch (step E of Figure 2.1). This step is replicated per each data pair and is

computationally demanding due to the applied resampling method based on FFTs. On the contrary, no particular storage is required since only a patch of the master and slave data pair is considered and the output is a text file containing the computed shift. Once these shifts are calculated for the entire dataset, they are solved for a single master (the reference one) and the images are resampled consequently.

Before moving to the interferogram generation, the evaluation of some parameters (as, for instance, the baseline values at each azimuth line and range sample per each interferometric pair) useful for the subsequent steps are carried out (block F of Figure 2.1). Having all the co-registered images and the DEM referred to a common geometry permits performing the subsequent differential interferogram and spatial coherence generation step (block G of Figure 3.1). For a given image pair, the considered step receives as input both the corresponding co-registered SLCs and the range files to simultaneously remove flat-Earth and topography contributions from the interferometric phase. This step just carries out the product, on a pixel basis, between the slave conjugate and the master, after applying a spectral shift filtering [32]; the synthetic interferograms generated from the range files are then removed. These operations are carried out at the sensor full spatial resolution and in the complex domain, thus implying high memory consumption. The option to apply a Goldstein's filtering [33], [34] is also available; these last steps require an intense use of two-dimensional (2-D) FFTs procedure. Moreover, the implementation of the Goldstein's filter is highly demanding in terms of disk storage and I/O data flow. It is worth noting that a multilook operation (for interferogram and coherence map generation) is carried out solely at the end of this step, thus permitting to reduce the size of the final output, not of the intermediate products.

The modulo- 2π restricted phase of each computed interferogram needs afterward to be unwrapped to retrieve the original phase [28]. This procedure is carried out by applying the extended minimum cost flow (EMCF) phase unwrapping (PhU) algorithm [35]. Although the EMCF approach considers temporal (temporal PhU) and spatial (range-azimuth) domain (spatial PhU) processing as conceptually

independent steps [35], the corresponding optimal sequential implementation, in terms of computational speed, shows some flow dependencies. In Figure 2.1, blocks I and H pertain to the PhU processing.

The phase unwrapping step is one of the most demanding in terms of memory consumption and computing power, particularly the spatial one. Indeed, it deals with wrapped and unwrapped interferogram stacks [three-dimensional (3-D) matrices] and iteratively retrieves the MCF solution on each interferogram for different network parameters. A final weighted combination of all the computed solutions, trying to maximize the temporal coherence [35] is also performed, implying again to deal with large data stacks and then memory resources.

Later, a pixel-based inversion of the unwrapped phase system of equation is carried out (step J of Figure 2.1), thus implying to exploit stacks of data, which require large memory availability. This allows us to retrieve the final displacement time-series; moreover, the residual topography is estimated by using an iterative approach (roughly 100 iteration per coherent pixel) thus having an important impact in the computational burden.

Note that, after retrieving the displacement time-series, it is also feasible to estimate and remove possible residual orbital phase ramps from the unwrapped interferograms (step K of Figure 2.1). This step is not particularly critical; however, for helping the PhU algorithm, it is better removing the estimated orbital phase ramps from the wrapped interferograms, implying that another PhU step on the "orbital error free" interferograms has to be performed (second run of step H, I, and J of Figure 2.1).

After performing temporal coherence estimation [35] and coherent point selection, block L provides the final displacement time-series.

As a final remark, it is worth noting that the atmospheric phase screen estimation, together with its removal, has not been included in the present analysis because in the SBAS chain, it is considered as a post-processing procedure [6] that can be performed following different.

2.2 Parallelization Rationale

In order to design the parallel version of the SBAS chain, a first fundamental aspect to be taken into account regards the opportunity of exploiting different parallel computing environments (i.e., cluster, grid, and emerging CC platforms) [13]–[20]. A special attention is given to the compatibility of the parallel SBAS chain with the mentioned CC paradigm. This issue certainly poses constraints on the design stage, which might conflict with the goal of achieving the parallel computation maximum efficiency. As an example, within a CC environment, the task partitioning should be carried out allowing the distribution of the computational load among cloud servers with minimal intertask communication. This implies the adoption of a suitable partitioning strategy with a certain granularity to coarsely distribute the work among several machines with limited communication overhead. Moreover, the message passing paradigm (e.g., de facto standard MPI) [38], [39], widely used for developing parallel applications, permits to achieve high performances by exploiting sophisticated parallel arrangements. However, its adoption could be problematic since the relevant application can turn out inconsistent within the mentioned CC environments [40].

A second important point to be taken into account concerns the development costs to carry out the parallel solution. High-performance parallel application fulfillment inevitably implies the development of appropriate algorithms specifically ad- dressed to the efficient exploitation of the available architectures (multi-core and multi-node clusters). Hence, the SBAS chain parallelization would require redesigning a significant part of the sequential version to take advantage of the modern parallel architectures. Nonetheless, easy implementation and rapid prototyping issues have to be considered, so that the development of the parallel model would be confined into workable time scales and not extremely heavy. Furthermore, the applied parallelization approach should preserve as much as possible the numerical precision of the sequential SBAS chain algorithms.

In general, devising a parallel computational model for a complex processing chain is a challenging task, since several aspects (data dependencies, task partitioning, inherent granularity, scheduling policy, load unbalancing, etc.) must be carefully taken into account during the development process, in order to finally achieve a parallel solution exhibiting significant efficiency and scalability [38], [39]. More specifically, it is worth emphasizing that, due to the heterogeneous nature of the algorithms comprised within SBAS processing chain, it is neither easy nor convenient to adopt a single parallelization strategy for all the steps of the chain. Accordingly, appropriate parallelization strategies have been adopted depending on the algorithmic structure of the particular processing step being evaluated (e.g., raw data focusing, image co-registration, phase unwrapping), as discussed in detail in the following.

In addition, the currently available hierarchical architectures (e.g., cluster) embrace multicore shared-memory computing nodes that are coupled via high-speed interconnections [38], [39]. In order to take advantage from such architectures, a dual-level parallelization strategy is required [38], [41]. Hence, multithreading-based shared-memory parallelization has also been exploited when possible and convenient (again in terms of cost–benefit ratio) to properly parallelize at a finer level on a single node, thus embracing both coarse-grain and fine-grain parallelism.

In the following, a high-level discussion aimed at clarifying the rationale of the parallel solution is provided.

In particular, two parallelization levels have been considered, i.e., the process and the thread level. The first one considers a coarse/medium granularity-based approach (mainly applied to the whole processing chain), the second one relies on a fine-grained parallelization and has been implemented only for the heaviest computational steps (A, H, and I of Figure 2.1)

Depending on the algorithmic structure of the considered step, the different parallelization strategies that have been adopted are classified according to the applied parallel methodology.

2.2.1 Process-Level Parallelization

In this section, the P-SBAS coarse/medium-grained parallelization strategy, addressed to the exploitation of multiprocessor systems with distributed memory, is presented. Figure 2.1 provides a synoptic representation of the proposed computational solution, which schematically describes the modular structure relevant to the process-level parallelism. In Figure 2.1, red blocks indicate parallel-performed processing steps and black blocks refer to sequential computations.

Within this scheme, different modules involve different parallelization approaches discussed in the following. Essentially, three main parallelization strategies have been applied according to the specific part of the processing chain.

1) *Pleasingly Parallel Processing Steps*: The term pleasingly parallel (or embarrassingly parallel) indicates computations that can be parallelized by requiring a minimal effort to partition the application into independent parallel parts [38], [39]. Such a kind of elaborations essentially exhibits minimal dependencies in terms of data, synchronization, or ordering.

   The elaboration steps A, C, E, and G of the SBAS elaboration chain (see Figure 2.1) are good candidates for being cast in this framework. In fact, these elaboration steps basically work on 2-D data, although the relevant algorithmic structure of the realized functional can be rather complex. For instance, raw data focusing (block A) applies to a collection of SAR raw data, whose sizes can typically ranges from a few dozens to several hundreds. Each elaboration acting on a single raw data can be then treated independently, leading to a set of independent computations concurrently running on different data, which can be naturally distributed to multiple processors. Similar considerations apply for the subsequent SAR image geometric co-registration (block C), image co-registration refining (block E), and interferogram generation (block G) processing steps.

The main drawback of this strategy concerns the limited scalability. However, such a limitation results reasonable when large SAR dataset, with respect to the number of nodes, are considered. The advantages in terms of simplicity of realization and of reduction of communication costs are, on the contrary, evident.

Moreover, it is worth noting that this straightforward parallel strategy adopts a data-centered approach to parallel computations, which is frequently used within the current CC technologies [40].

2) *Medium-Grained Data Parallelism*: The term data parallelism [38], [39] basically refers to a data manipulation performed in parallel, by exploiting multiple processors that work on different portions of the data. This approach relies on the fact that the same operation must be applied to different elements of a large data structure. Such a kind of parallelization has been specifically exploited within the elaboration steps B and J of the SBAS processing chain (Figure 3.1), both dealing with the processing of large multidimensional (2-D and 3-D) arrays that can be performed by independently executing the same elaboration on pieces of the input data.

   More specifically, in the processing step B (see Figure 2.1), which concerns the conversion of the DEM (pertinent to the area of interest) in SAR coordinates, the input 2-D array is partitioned in 2-D subarrays of equal size to be distributed among homogeneous nodes, so that the load balancing is easily achieved. The 2-D subarrays are then distributed in chunks so that each processor is responsible for the evaluation of several subarrays; a static scheduling is assumed. Finally, a proper combination of the parallel processed portions is implemented. The size of the 2-D subarrays is selected according to the trade-off between the single subarray processing time and the minimization of the associated communications cost. The granularity level of the proposed solution depends on the number of subarrays in which the DEM is split. This number

depends on the extent of the scene observed by the SAR sensor and on the relevant SAR image resolution.

Within the processing step J (see Figure 2.1), a similar parallel approach is adopted to deal with the pixel-based inversion of the unwrapped phase system [6]. This operation acts on a large 3-D array (whose dimensions are azimuth, range and interferogram list) containing the overall unwrapped interferometric phases. On the third dimension (interferogram list) of this large 3-D array, i.e., on a pixel basis, a sequence of operation is carried out for finally retrieving the ground displacement time-series. These operations are mainly related to residual topography estimation and removal [6], and to the unwrapped phase system solution via the singular-value decomposition (SVD) method. The considered 3-D array can be then partitioned in equal size of one- dimensional (1-D) subarrays (whose computation is carried-out independently) to be distributed among homogeneous nodes, so load balancing is easily achieved. The 1-D subarrays are distributed in chunks, so that each processor is responsible for evaluation of several 1-D subarrays. Also in this case, a static scheduling is considered. We emphasize that the total number of 1-D subarrays to be processed strongly depends on the analyzed scenario, i.e., on the number of considered pixels. Nonetheless, for ENVISAT scenario, it can be on average comprised within the range $10^4 - 10^6$ pixels. We further note that, in this case, the considered approach could also be amenable of a dual-level [38], [41] parallelization arrangement that, however, has not yet been implemented.

3) *Coarse-Grained Parallelism by Functional Decoupling*: In this section, we discuss an ad hoc strategy to cope with a parallel solution for the phase unwrapping problem [35]. It is worth highlighting that the relatively simple aforementioned strategies apply only when the relevant applications concern problems with a regular structure. On the contrary, we stress that for the elaborations including more involved computations (as in the case of the phase- unwrapping

problem), it is not generally possible to directly obtain independent calculations that can be solved in parallel.

We note that the considered processing steps (H and I) pertaining to the phase unwrapping problem are the most time-consuming ones, so strongly motivating the development of an ad hoc parallel algorithm. Indeed, the overall unwrapping process might be extremely demanding, particularly for long interferogram sequences.

To clarify the logic of the specific parallelization strategy, preliminary considerations on the algorithm structure of the optimized sequential version and the inherent subtle flow dependencies are in order.

The EMCF formulation [35] is an extension of the MCF technique [42]. Specifically, EMCF also exploits the temporal relationships among different multitemporal interferograms, thus improving the MCF technique. We recall that, according to the EMCF optimized sequential version, the temporal PhU is iteratively evaluated for different $(\Delta z, \Delta v)$pairs, where Δz and Δv stand, respectively, for the searching domains relevant to the error in the knowledge of the scene topography and the deformation velocity variations, along the considered spatial arc [35]. For each $(\Delta z, \Delta v)$ pair, the result of the temporal PhU operation is then used to bootstrap the associated spatial PhU operation, which is performed on each single interferogram through the application of the basic MCF unwrapping technique.

At this point, it is crucial to recognize that an intrinsic flow dependency exists, since the results of the temporal PhU obtained for a certain $(\Delta z, \Delta v)$ pair are obtained by updating the temporal PhU results relevant to the previous evaluated $(\Delta z, \Delta v)$ pairs. Therefore, this flow dependency, which is particularly effective in the sequential version, results to be not directly eliminable and hampers concurrent computations of the elaboration pertinent to different $(\Delta z, \Delta v)$ pairs. To overcome such a limitation, we rely on a proper functional decoupling of the computational

problem. The aim is then, by splitting the loop into serial and parallel portions, to identify a suitable decomposition of the problem into sub-problems that in part can be solved concurrently. Accordingly, we rearrange the relevant algorithm structure in order to obtain spatial and temporal PhU disjoint computations. Thus, we first sequentially compute the intermediate results obtained by applying the temporal PhU to all successive $(\Delta z, \Delta v)$ pairs (processing step H). Once all these intermediate results are available, the spatial PhU computation can be carried out for each $(\Delta z, \Delta v)$ pair independently, so enabling parallel computation of the spatial PhU of the different $(\Delta z, \Delta v)$ pairs (Block I of Figure 2.1). On the contrary, the sequence of temporal PhU operation remains to be processed sequentially (Block H of Figure 2.1).

Therefore, the adopted decomposition is effective, since the entire spatial PhU problem is reduced to the solution of independent sub-problems, pertinent to different $(\cdot, \Delta v)$ pairs, which can be executed concurrently in a coarse-grain fashion. The inherent coarse-grained parallelism is finally implemented by distributing the individual sub-problems to participating processors in a round-robin fashion. Also in this case, we adopt a static scheduling, with minimum communication and synchronization cost.

2.2.2 Thread-Level Parallelization

In order to complement the coarse/medium-grained parallelization adopted at process level, we also employ fine-grained shared-memory parallelization strategy. Specifically, this approach primarily concerns the blocks A, H, and I, since (spatial and temporal) phase unwrapping computation and raw data focusing are the most time-consuming calculations.

An exhaustive discussion of the multithreading parallelization strategy requires a detailed algorithm structure analysis, whose description is beyond the scope of this thesis. However, a brief description of the focusing step thread-level parallelization is given, hereafter, to clarify the employed rationale.

The used approach consists in processing independent portions of the input data in parallel on multiple cores according to the data parallelism strategy through multithreads programming. In particular, the range-compressed data are split into overlapping azimuth blocks which are processed in parallel on multiple cores by exploiting multithread programming, first to estimate the Doppler centroid frequency [28] and then to retrieve the azimuth compressed signal. The number of blocks is defined in a dynamic way as explained in the following; it depends on the raw data dimension, the number of available cores, and the theoretical constraints for an accurate processing. Considering that, for a correct focusing, the azimuth blocks size has a lower limit equal to the number of pixels corresponding to two synthetic antennas and an upper limit that depends on the Doppler frequency variation along the azimuth direction, the dimension of the blocks is chosen by considering all the possible power of 2, 3 and 5 dimensions (for a FFT execution) between the above- mentioned limits, as well as the number of available cores. To guarantee the load balancing, among the possible ones, the algorithm sets the block size providing a number of blocks that optimizes the exploitation of the available cores. The elaborated blocks have got all the same size; an overlap area between different blocks is also considered. Besides, the fact that the blocks have all the same size preserves the load balancing as well. Finally, the focused blocks are appropriately gathered.

The granularity level is dictated by the minimum allowed dimension of the azimuth blocks in which the raw data can be split, thus being bound by a theoretical constraint.

2.2.3 Implementation Details

Algorithms constituting the P-SBAS chain have been mainly implemented by using interactive data language (IDL) version 8.2, characterized by a large diffusion in the scientific community. The computational heaviest parts have been implemented in Fortran or C. It is worth noting that the logic of the proposed P-SBAS scheme is independent from the selected implementation language.

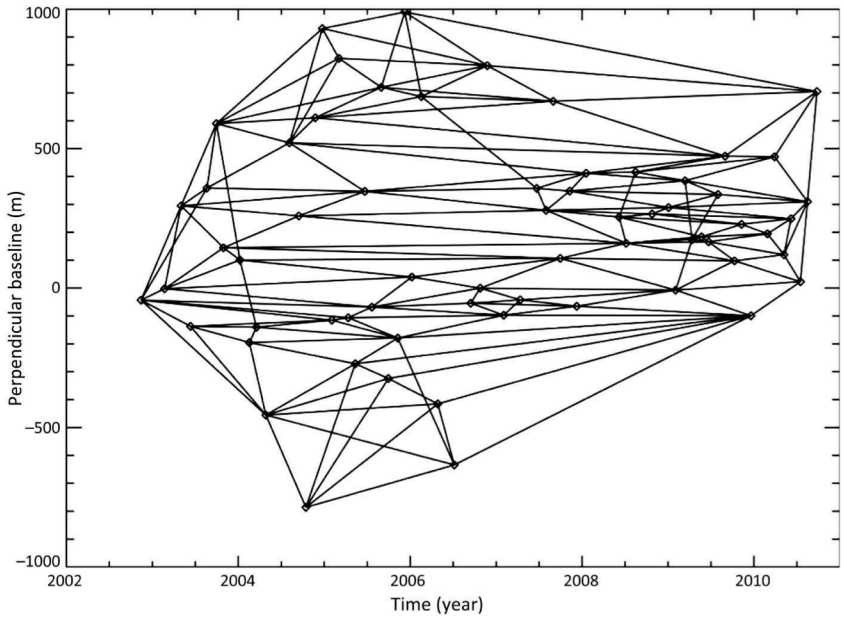
Regarding the process-level parallelization, the management, scheduling, and synchronization of the parallel tasks on different nodes of the cluster have been implemented by using UNIX/Linux bash scripts. This is a reasonable choice as it matches well with the parallelization scheme adopted for the primary level and also required a limited development cost. A different approach was used for the thread-level parallelization: both the IDL_IDLBridge object class, in the case of IDL code, and the OpenMP [41] application programming interface, in the case of Fortran routines, have been exploited.

2.3 Experimental Results

This section is devoted to present the experimental results obtained by applying the proposed P-SBAS processing scheme to a realistic case study. In particular, a performance analysis of the implemented prototype has been carried out to experimentally demonstrate its computational effectiveness through widely used metrics. The experimental analysis is carried out on a specific computational infrastructure. Moreover, the obtained results are discussed and some considerations are drawn with respect to the applicability of the proposed solution to scenarios of interest.

2.3.1 Case Study Description

A test site, around the Napoli Bay, a volcanic and densely urbanized area in Southern Italy including the active caldera of Campi Flegrei, the Vesuvio vulcano, and the city of Naples, is considered. We employ the full available ASAR-ENVISAT archive, composed by 64 scenes, spanning the 2002–2010 time interval and acquired from ascending orbits, which represents a typical ENVISAT data stack. This dataset has been processed by using the P-SBAS algorithm to generate the deformation time-series. In particular, 195 differential interferograms have been produced, characterized by a (perpendicular) baseline value smaller than 400 m and a maximum temporal separation of about 1500 days (see Figure 2.2). The overall analysis has been carried out on DInSAR products obtained following a complex multilook operation with 4 looks in the range direction and 20 in the azimuth one; hence, the pixel dimension is about $90 \times 90\ m$ on the ground.



Napoli Bay area	
Sensor	ENVISAT
Area	100 x 100 km²
# Images	64
# Full resolution pixels	18000 x 5000
# Interf	195
# Unwrapped points	97174

Figure 2.2: Distribution of the employed SAR dataset in the temporal/perpendicular baseline plane, with the relevant characteristics in the box on the right.

Precise satellite ENVISAT orbital information and the 3-arcsecond SRTM DEM of the area have been used. In order to provide an overall picture of the achieved results, we present, in Figure 2.3, the

estimated mean deformation velocity obtained by applying P-SBAS to the considered case study. Note that this map is represented in SAR coordinates and then superimposed on a SAR amplitude image of the investigated area.

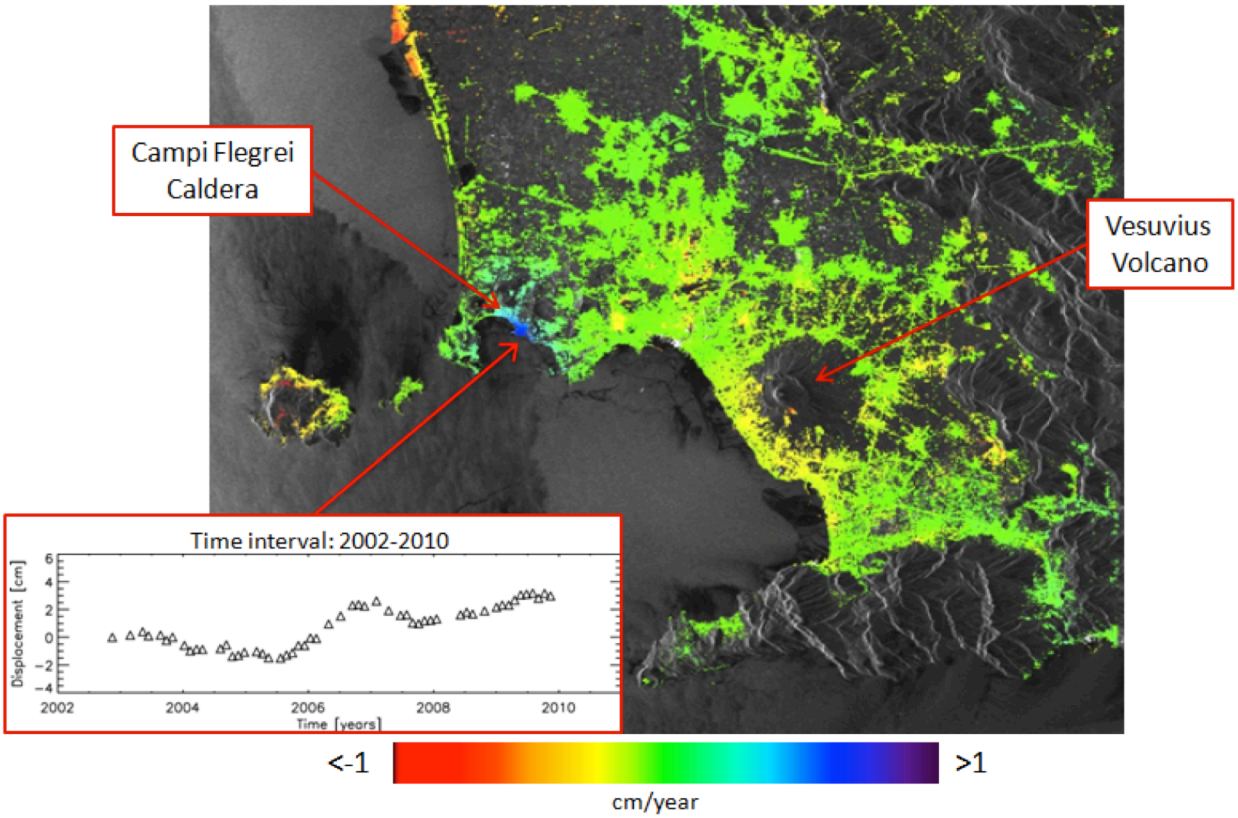


Figure 2.3: Mean deformation velocity map of the Napoli Bay area, obtained by applying the proposed P-SBAS algorithm. The graph of the displacement time series relevant to a pixel located in the area of maximum deformation is also shown.

The estimated mean deformation velocity has been computed in coherent areas only; accordingly, areas in which the measurement accuracy is affected by decorrelation noise have been excluded from the false-color map. In particular, we emphasize that in this representation (Figure 2.3), a remarkable deformation pattern corresponding to the area of the Campi Flegrei caldera is evident. Note also that, for each coherent point of the scene, the computation of the temporal evolution of the detected

deformation has been carried out. For instance, the chronological sequence of the computed displacement of a specific point (located in the maximum deforming area of the Campi Flegrei caldera) is shown in the box of Figure 2.3.

As a final remark, we stress that the P-SBAS scheme has been carefully designed to preserve the result accuracy. We also benefit from the nature of some algorithms of the processing chain, which filter out possible numerical errors, i.e., the phase unwrapping steps that involve the solution of an integer network flow problem [42]. Accordingly, the comparison between the results obtained with the sequential and parallel version shows that the difference is negligible.

2.3.2 Computational Platform

The architecture used to carry out our experimental analysis is made of 8 nodes, each one is equipped with 2 quad-core no-hyperthreaded CPU (AMD Opteron 2356-2.2 GHz) and 32 GB of RAM (see Table 2.1). The cluster has a shared storage implemented through a network file system (NFS) with 1 Gbit/s network bandwidth and 300 MB/s server disk bandwidth.

2.3.3 Performance Metrics

In this section, we first introduce two performance measures (metrics): these are the *speedup factor* and the *efficiency*. Then, we focus on the load unbalancing and the presence of a sequential fraction of the parallel algorithm, which constitute two primary sources of inefficiency.

TABLE 2.1
REFERENCE COMPUTATIONAL PLATFORM CONFIGURATION

\	Configuration Information
Operative System	GNU/Linux 3.2.5
Processor	2 quad-core CPU (*AMD® Opteron®* 2356-2.2 GHz)
Number of nodes	8
Memory Size per node	32 GB of RAM
Interconnection	1 Gb Ethernet

In order to quantitatively appreciate the benefit of parallelism achievable with the proposed solution, we resort to the notion of speedup, which compares the resulting parallel execution time with the sequential execution time on one processor [38], [39]. Accordingly, the speedup of a parallel program with parallel execution time T_N is defined as:

$$S_N = \frac{T_1}{T_N} \tag{1}$$

where N is the number of processors used to solve a problem and T_1 is the execution time of the sequential implementation to solve the same problem. Note that an upper bound for the speedup exists, thus the following inequality $S_N \leq N$ holds. Accordingly, due to several inefficiencies, parallel linear speedup ($S_N = N$) is achievable only in ideal conditions. An alternative performance measure for a parallel program is the efficiency [38], [39]:

$$\varepsilon = \frac{S_N}{N} \tag{2}$$

The efficiency is a measure of the speedup achieved per processor. It should be noted that an ideal speedup corresponds to a unitary efficiency. Primary sources of inefficiency that may inevitably limit the degree of parallelism are now addressed.

First, the unbalanced load affects parallelization efficiency, since it causes some resources to be underutilized, so hampering the performances. A pertinent quantitative description can be given in terms of the *average imbalance* of the load β defined as [38], [39]

$$\beta = \frac{total\ size\ of\ all\ subtasks}{N \cdot (size\ of\ largest\ subtask)} \tag{3}$$

It should be noted that when all the subtasks are perfectly balanced we get $\beta = 1$. Roughly speaking, a parallel computa- tion is well balanced if all processors perform the same amount of computation. On the contrary, when the entire task is performed by a single processor, we have $\beta = {}^{1}/_{N}$.

Second, the presence of data dependencies may inevitably limit the degree of parallelism. Indeed, real algorithms contain some (serial) parts that cannot be divided among processors. In this respect, the effect on the attainable speedup can be captured quantitatively by the well-known *Amdahl's law* [38], [39]

$$S_N = \frac{1}{f_s + \frac{1 - f_s}{N}} \qquad with\ 0 \le f_s \le 1 \tag{4}$$

where f_s is the fraction of a parallel program that must be executed sequentially. It is worth mentioning that the simple formulation (Equation 4) of *Amdahl's law* does not take into account the load unbalancing. Finally, we recall that another important source of inefficiency is ascribable to communication overhead.

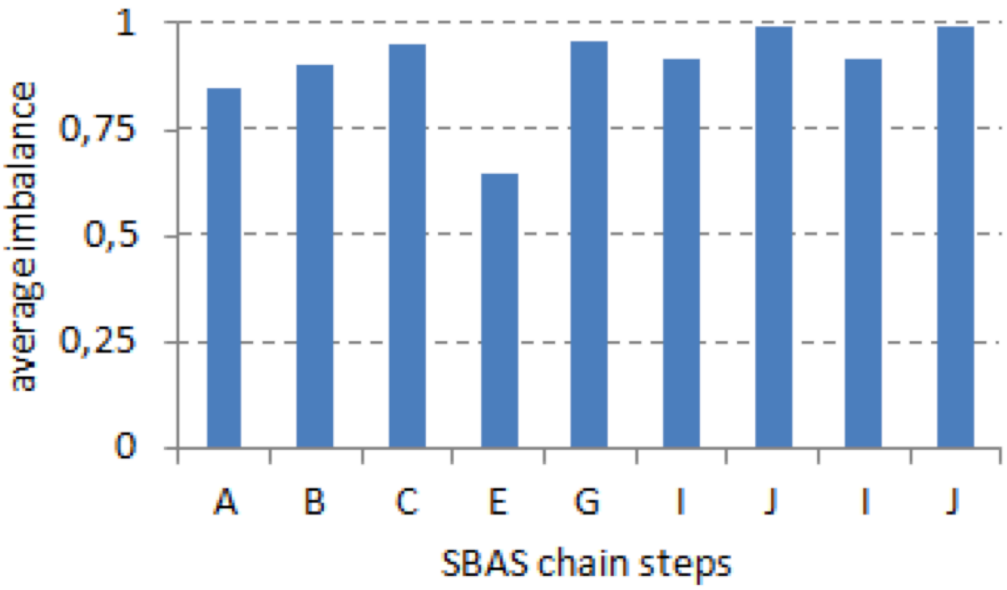


Figure 2.4: Distribution of β as a function of the relevant parallel processing step, obtained with 16 processors. Note that steps I and J are present twice since the processing has repeated two times due to orbital ramps removal.

2.3.4 Performance Analysis

In this section, we present some experimental results relevant to the case study considered in paragraph 2.3, the performance analysis has been carried out on the cluster described in paragraph 2.3.2. Scalability with the number of engaged processors is also addressed.

Concerning the considered case study, the associated computing time of the sequential SBAS version is of about 125 hours (5.2 days), while the computation performed with P-SBAS engaging 16 processors requires roughly 17 hours. In addition, the inherent fraction of sequential part, for the considered dataset, was estimated to be 5% ($f_s = 0.05$).

First of all, we analyze the average imbalance factor β relevant to each processing step. In Figure 2.4, the distribution of β for each parallel processed step obtained with $N = 1$ is depicted. As expected, the static scheduling strategy we adopt does not permit a perfect load balancing. Except for the processing step E, which is amenable of further improvements, we emphasize that the different parallel processing steps exhibit a load distribution that, however, results to be reasonably balanced.

A speedup analysis, aimed at quantifying the parallelism benefits of the proposed P-SBAS solution with respect to the sequential version, has been subsequently carried out. First, we focus on the speedup evaluated with reference to a specific processing step only. Accordingly, Figure 2.5 - 2.7 depict the experimental speedup (blue line/diamonds) relevant to the A, B, and I processing steps (i.e., the raw data focusing, DEM conversion in SAR coordinates, spatial phase-unwrapping), respectively (see also Figure 2.1). Note that, as a reference, the ideal achievable linear speedup (red line/squares) is also shown. Similar evaluations have been carried out for the other processing steps; however, for the sake of brevity, they are not shown. The trend in Figure 2.5 shows that, as the number N of the engaged processors increases, the discrepancy, between the ideal linear speedup and the actual one, increases. This kind of behavior is clearly explained by the fact that the average imbalance factor decreases as N increases, assuming that the associated communication overhead is negligible. Indeed, since the time to process diverse raw data might be not exactly the same, due to the size of the input data that can be different, the number of raw data to be focused per processor reduces with N, with consequent deterioration of the resulting load balancing. It should be noted that the load imbalance among processors we experience for the point A (see also Figure 2.4 relevant to the case $N = 16$) is also

closely connected with the adopted design strategy involving a coarse granularity. The relevant performances are, however, satisfactory. In Figure 2.6, the speedup of the processing step B, DEM conversion in SAR coordinates is presented; according to the value of the corresponding average imbalance factor, it shows a quasi-linear behavior up to 16 processors. Concerning the parallel computation of the spatial phase unwrapping step, as it clearly appears in the speedup graph (Figure 2.7), a near linear scaling is exhibited up to 16 processors, so showing a good scalability. Also, in this case, such a behavior is in good agreement with the value of the average imbalance factor pertinent to the point I (see also Figure 2.4 relevant to the case $N = 16$).

Finally, the speedup of the overall P-SBAS processing chain is evaluated. In Figure 2.8, this speedup is depicted as a function of the number of processors N (blue line/diamonds). As a reference, the ideal achievable linear speedup (red line/squares) and speedup predicted by the *Amdahl's law* (green line/triangles) with $f_s = 5\%$ are also shown. In conclusion, the parallel solution proposed in this paper permits to obtain satisfactory performance and scalability, thus achieving a significant reduction of the overall elaboration time with crucial repercussion on application scenario.

2.3.5 Computational Remarks

It is important to emphasize that in the case in which we were able to attain a perfect balancing of the workload for each parallel processing step, the experimental trend would become practically superimposed on the relevant Amdahl's law (see green line/triangles of Figure 2.8). This observation is important to understand the relevance of the effect of the load unbalancing on the achieved speedup. It should also be noted that the discrepancy between the experimental results and the Amdahl's law is, however, smaller than the discrepancy between the ideal linear speedup and the Amdahl's law. Accordingly, as far as load balancing is concerned, the most immediate feature to note is that the static

scheduling relevant to the adopted first-level parallelization strategy is not the major limit to the parallelism, as long as a small/medium cluster is concerned. This clearly demonstrates the effectiveness of the presented parallelization strategy, showing good speedups and good scalability on small/medium clusters.

On the contrary, the major source of inefficiency for the adopted scheme is ascribable to the presence of a non-negligible intrinsic sequential fraction. Nonetheless, this major source of inefficiency, even if liable to a further reduction, considering the complex nature of the entire processing chain, remains essentially non-eliminable.

According to *Amdahl's law*, the maximum achievable speedup is dictated by the sequential fraction and in the analyzed case, with a moderate sequential portion of 5%, a maximum theoretical speedup equal to 20 can be achieved, regardless of the number of engaged processors (N). This has important implications on the maximum number of processors that can be effectively exploited.

However, the weight of the sequential fraction with respect to the parallel one is dependent on the considered dataset, and it is expected to decrease as the size of the considered dataset increases.

2.4 Unsupervised P-SBAS Processing Chain Implementation

In this section, we refer to the P-SBAS algorithm above described and on the unsupervised implementation of its processing chain.

The idea is to develop an algorithm capable to run, in a fully automated and unsupervised manner, the complex end-to-end P-SBAS processing chain, without significantly affecting the achieved result accuracy.

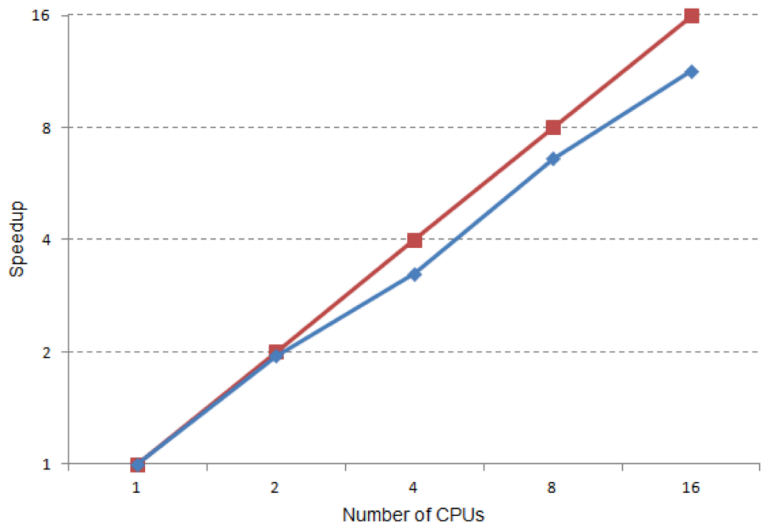


Figure 2.5: Raw data focusing—processing step A: speedup as a function of the number of processors N (in blue/diamonds). The ideal achievable speedup (red/ squares) is also shown.

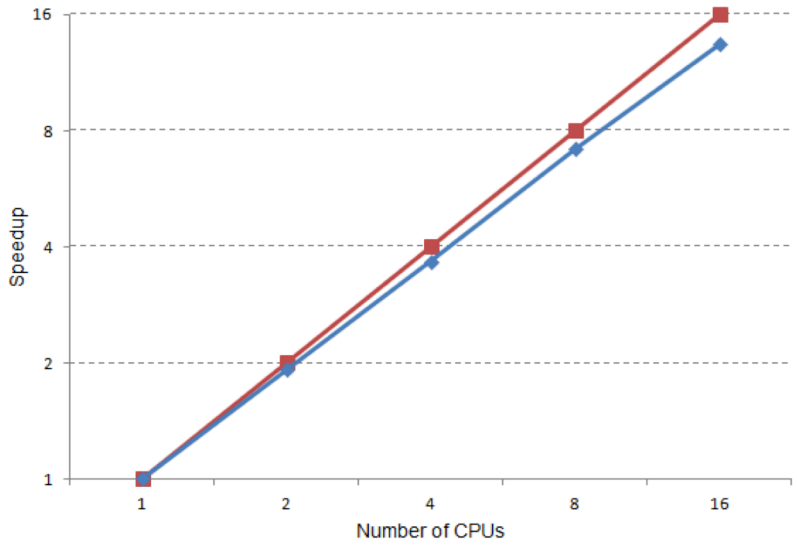


Figure 2.6: DEM conversion in SAR coordinates—processing step B: speedup as a function of the number of processors N (in blue/diamonds). The ideal achievable speedup (red/squares) is also shown

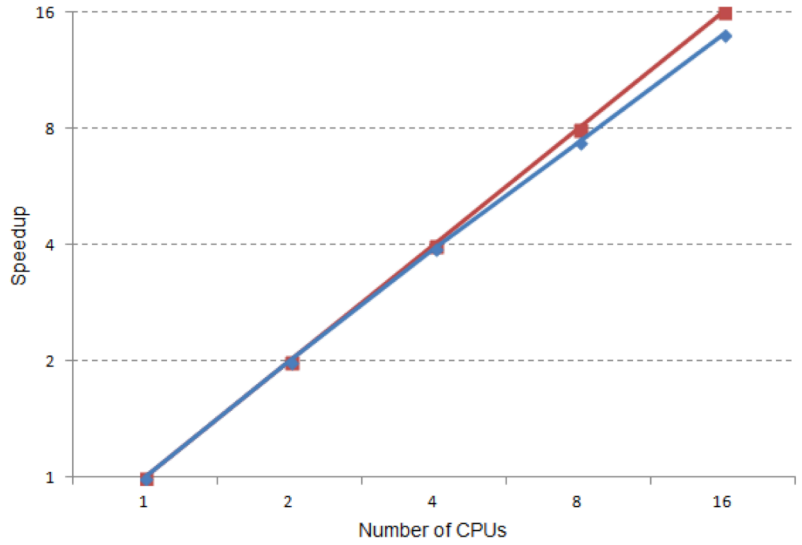


Figure 2.7: Spatial phase-unwrapping—processing step I: speedup as a function of the number of processors N (in blue/diamonds). The ideal achievable speedup (red/squares) is also shown.

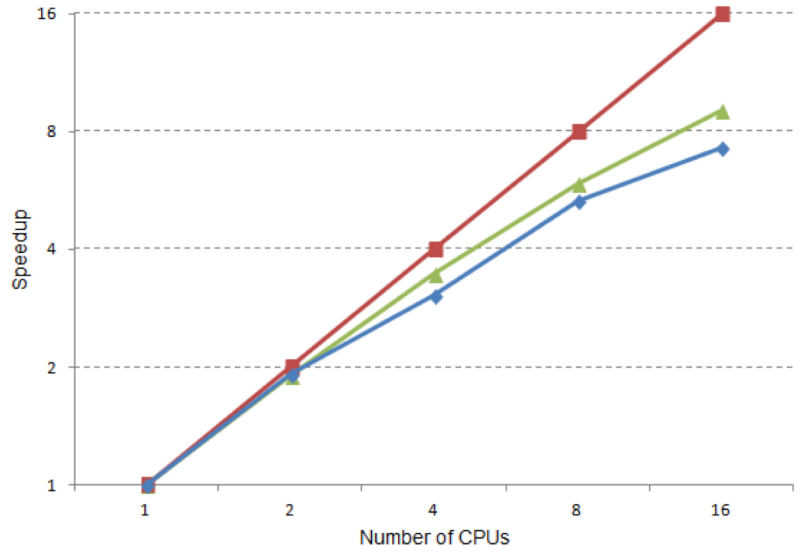


Figure 2.8: OverallP-SBAS processing chain: speedup as a function of the number of processors N (in blue/diamonds). The ideal achievable speedup (red/squares) and the Amdahl's law behavior (green/triangles) are also shown.

Multi-temporal DInSAR processing through the P-SBAS approach represents a rather complex task that implies the correct execution of several, subsequent steps to finally retrieve surface deformation mean velocity maps and the corresponding time series with accuracies of one millimeter per year and of few millimeters, respectively [3,7,27].

In general, DInSAR processing requires some skilled user interventions and evaluations to increase the quality and reliability of generated DInSAR results. In order to avoid user interaction and to release a P-SBAS processing chain able to run in a completely unsupervised way, several algorithms and routines have been developed, thus allowing us to strongly improve the P-SBAS chain also in terms of efficiency and robustness. Consequently, we implemented a P-SBAS processing chain capable to run starting from the SAR RAW data (Level 0) selection up to the generation of surface deformation time series. In particular, the unsupervised P-SBAS chain permits to automatically:

- Identify, within the input SAR dataset, data that are acquired at the same epoch but that partially cover the user's Area of Interest (AoI) and to evaluate the possibility to mosaic them for increasing the investigated area spatial coverage. This feature also permits to automatically discard SAR acquisitions that do not cover the selected AoI. Note that the mosaicking can be performed on adjacent frames of the same track, but not across different tracks.

- Identify and analyze only the portion of SAR data that cover the AoI, thus reducing the amount of data to be processed and, consequently, the elapsed computing time;

- Select, among the overall SAR acquisitions, the master image, which is used to identify the reference geometry to minimize the possible error sources during the DInSAR processing. The master image is selected as the closest to the barycenter of the SAR data acquisition distribution in the temporal/perpendicular baseline $T \times b_{\perp}$) plane (see Figure 1.9, Chapter 1);

- Improve the precision of the orbit metadata, e.g., the acquisition start time, by making use of DEM (Digital Elevation Model) information. Indeed, an incorrect knowledge of the acquisition start time basically translates in a rigid misalignment between the DEM and the SAR acquisitions. To calculate this residual rigid shift, the implemented algorithm first selects an area with high coherence and high phase gradient, within an interferogram less affected by deformation (short temporal baseline) and sensitive to the topography (i.e., with a relatively high perpendicular baseline). Then, it performs a cross-correlation between the phase gradients of the so identified interferogram and the synthetic topography phase computed from the DEM. The peak of the cross-correlation identifies the requested shift;

- Refine the user-selected reference point of the displacement time series by looking for the one that is affected by the lowest noise level (as defined in [45]) in a surrounding area, thus improving the degree of accuracy of the final DInSAR results. To this aim the algorithm performs, for each interferogram, the phase unwrapping procedure of a patch around the user selected reference point. The point with the higher temporal coherence is then identified as the new reference;

- Select the pixels to be unwrapped by properly identifying only those that are not significantly affected by noise, thus improving the quality of the final results, taking into account the spatial coherence of each pixels in each data pair;

- Evaluate the temporal coherence threshold, in dependence of the number of the exploited SAR acquisitions, according to an empirical relationship derived from an extensive analysis of ESA C-band SAR data.

2.5 Summary

In this paper, a parallel and automatic computing solution for the SBAS processing chain has been presented. This is particularly suitable to effectively exploit the widely available parallel hierarchal platforms and the huge SAR data archives collected by present and future SAR missions; accordingly, the proposed P-SBAS solution represents a highly valuable tool for the analysis of the complex phenomena characterizing the surface deformation dynamics of large areas of Earth.

Our parallel computing solution has been analyzed by using different metrics, thus demonstrating the effectiveness of the proposed approach. In particular, it turned out that the carried out task partitioning, inherent to the applied first-level parallelization, permits to obtain good performances even by employing a simple static scheduling, since the associated communication overhead results to be extremely reduced. Nonetheless, insofar small/medium clusters are considered, the scalability of the parallel SBAS chain has been successfully demonstrated. Limitations concerning the scalability have also been discussed. These issues pertain the arising unbalancing ascribable to the extremely coarse granularity involved and, more important, the intrinsic sequential part. These limitations are, however, reasonable within the considered application context. In addition, by taking advantage also of the employed fine-grained multithreading parallelization, the proposed P-SBAS solution is suitable for effectively exploiting computational platforms with several cores.

References

[1] Gabriel, R. Goldstein, and H. Zebker, "Mapping small elevation changes over large areas: Differential radar interferometry," J. Geophys. Res., vol. 94, no. B7, pp. 9183–9191, 1989.

[2] D. Massonnet et al., "The displacement field of the Landers earthquake mapped by radar interferometry," Nature, vol. 364, no. 6433, pp. 138–142, Jul. 1993.

[3] D. Massonnet, P. Briole, and A. Arnaud, "Deflation of Mount Etna monitored by spaceborne radar interferometry," Nature, vol. 375, no. 6532, pp. 567–570, Jun. 1995.

[4] S. Usai, "A least-squares database approach for SAR interferometric data," IEEE Trans. Geosci. Remote Sens., vol. 41, no. 4, pp. 753–760, Apr. 2003.

[5] A. Ferretti, C. Prati, and F. Rocca, "Permanent scatterers in SAR interfer- ometry," IEEE Trans. Geosci. Remote Sens., vol. 39, no. 1, pp. 8–20, Jan. 2001.

[6] P. Berardino, G. Fornaro, R. Lanari, and E. Sansosti, "A new algorithm for surface deformation monitoring based on small baseline differential SAR interferograms," IEEE Trans. Geosci. Remote Sens., vol. 40, no. 11, pp. 2375–2383, Nov. 2002.

[7] M. Bonano, M. Manunta, A. Pepe, L. Paglia, and R. Lanari, "From previous C-band to new X-band SAR systems: Assessment of the DInSAR mapping improvement for deformation time-series retrieval in urban areas," IEEE Trans. Geosci. Remote Sens., vol. 51, no. 4, pp. 1973–1984, Apr. 2013.

[8] A. Rucci, A. Ferretti, A. Monti Guarnieri, and F. Rocca, "Sentinel 1 SAR interferometry applications: The outlook for sub millimeter measurements," Remote Sens. Environ., vol. 120, pp. 156–163, May 2012.

[9] S. Salvi et al., "The Sentinel-1 mission for the improvement of the scientific understanding and the operational monitoring of the seismic cycle," Remote Sens. Environ., vol. 120, pp. 164–174, May 2012.

[10] L. Fusco and R. Cossu, "Past and future of ESA Earth observation grid," Mem. S.A. It., vol. 80, pp. 461–476, 2009.

[11] R. Lanari et al., "A small-baseline approach for investigating deformations on full-resolution differential SAR interferograms," IEEE Trans. Geosci. Remote Sens., vol. 42, no. 7, pp. 1377–1386, Jul. 2004.

[12] M. Bonano, M. Manunta, M. Marsella, and R. Lanari, "Long-term ERS/ENVISAT deformation time-series generation at full spatial resolu- tion via the extended SBAS technique," Int. J. Remote Sens., vol. 33, pp. 4756–4783, Feb. 2012.

[13] C. A. Lee, S. D. Gasster, A. Plaza, C.-I. Chang, and B. Huang, "Recent developments in high performance computing for remote sensing: A review," IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 4, no. 3, pp. 508–527, Sep. 2011.

[14] N. Sadashiv and S. M. D. Kumar, "Cluster, grid and cloud computing: A detailed comparison," in Proc. 6th Int. Conf. Comput. Sci. Edu. (ICCSE), Aug. 2011, pp. 477–482.

[15] R. Prodan and T. Fahringer, Grid Computing, Experiment Management, Tool Integration, and Scientific Workflow. New York, NY, USA: Springer- Verlag, 2007.

[16] K. H. Hoffmann and A. Meyer, Eds., Parallel Algorithms and Cluster Computing: Implementations, Algorithms and Applications. New York, NY, USA: Springer-Verlag, 2006.

[17] M. Cafaro and G. Aloisio, Eds., Grids, Clouds and Virtualization. New York, NY, USA: Springer-Verlag, 2011.

[18] F.Gebali, Algorithms and Parallel Computing. Hoboken,NJ,USA:Wiley, 2011.

[19] T.Rauber and G.Runger, Parallel Programming for Multicore and Cluster Systems. New York, NY, USA: Springer-Verlag, 2010.

[20] F. Travostino, J. Mambretti, and G. Karmous-Edwards, Grid Networks, Enabling Grids With Advanced Communication Technology. Hoboken, NJ, USA: Wiley, 2006.

[21] Y. Ma et al., "Generic parallel programming for massive remote sensing data processing," in Proc. IEEE Int. Conf. Cluster Comput. (CLUSTER'12), Sep. 2012, pp. 420–428.

[22] C. Jiang, X. Xu, J. Wan, J. Zhang, and Y. Li, "Java multi threaded based parallel remote sensing image interpretation in desktop grid," in Proc. 5th Annu. ChinaGrid Conf. (ChinaGrid), Jul. 2010, pp. 51–59.

[23] F. Kong, J. Zhao, and B. Yue, "Research on parallel processing of SAR imaging algorithm," in Proc. 2nd Asian-Pacific Conf. Synth. Aperture Radar (APSAR'09), Oct. 2009, pp. 784–787.

[24] M. Malanowski et al. "Real-time high-resolution SAR processor using CUDA technology," in Proc. 14th Int. Radar Symp. (IRS'13), Jun. 2013, pp. 673–678.

[25] H. Zhou, X. Yang, H. Liu, and Y. Tang, "Research on CUDA-based SIFT registration of SAR image," in Proc. Int. Conf. Parallel Process. (ICPP'05), Jun. 2005, pp. 129–136.

[26] W. Chapman et al., "Parallel processing techniques for the processing of synthetic aperture radar data on GPUs," in Proc. IEEE Int. Symp. Signal Process. Inf. Technol., 2011, pp. 573–580.

[27] R. Lanari et al., "An overview of the small baseline subset algorithm: A DInSAR technique for surface deformation analysis," Pure Appl. Geophys. (PAGEOPH), vol. 164, no. 4, pp. 637–661, Apr. 2007.

[28] G. Franceschetti and R. Lanari, Synthetic Aperture Radar Processing. Boca Raton, FL, USA: CRC Press, 1999.

[29] J. C. Curlander and R. McDonough, Synthetic Aperture Radar-System and Signal Processing. Hoboken, NJ, USA: Wiley, 1991.

[30] E. Sansosti, P. Berardino, M. Manunta, F. Serafino, and G. Fornaro, "Geometrical SAR image registration," IEEE Trans. Geosci. Remote Sens., vol. 44, no. 10, pp. 2861–2870, Oct. 2006.

[31] J. J. Knab, "The sampling window," IEEE Trans. Inf. Theory, vol. 29, no. 1, pp. 157–159, Jan. 1983.

[32] F. Gatelli et al., "The wave-number shift in SAR interferometry," IEEE Trans. Geosci. Remote Sens., vol. 32, no. 4, pp. 855–865, Jul. 1994.

[33] R. M. Goldstein and C. L. Werner, "Radar interferogram filtering for geophysical applications," Geophys. Res. Lett., vol. 25, no. 21, pp. 4035–4038, 1998.

[34] P. A. Rosen, S. Hensley, G. Peltzer, and M. Simons, "Updated repeat orbit interferometry package released," EOS, vol. 85, no. 5, p. 47, Feb. 2004.

[35] A. Pepe and R. Lanari, "On the extension of the minimum cost flow algorithm for phase unwrapping of multitemporal differential SAR interferograms," IEEE Trans. Geosci. Remote Sens., vol. 44, no. 9, pp. 2374–2383, Sep. 2006.

[36] F. Onn and H. A. Zebker, "Correction for interferometric synthetic aperture radar atmospheric phase artifacts using time series of zenith wet delay observations from a GPS network," J. Geophys. Res., vol. 111, no. B09102, pp. 1–16, 2006.

[37] R. Jolivet, R. Grandin, C. Lasserre, M. P. Doin, and G. Peltzer, "Systematic InSAR tropospheric phase delay corrections from global meteorological reanalysis data," Geophys. Res. Lett., vol. 38, no. 17, pp. 1–6, 2011.

[38] G. Hager and G. Wellein, Introduction to High Performance Computing for Scientists and Engineers. Boca Raton, FL, USA: CRC Press, 2010.

[39] H. El-Rewini and M. Abd-El-Barr, Advanced Computer Architecture and Parallel Processing. Hoboken, NJ, USA: Wiley, 2005.

[40] A. Goscinski, R. Buyya, and J. Broberg, Eds., Cloud Computing: Principles and Paradigms, Hoboken, NJ, USA: Wiley, 2011.

[41] B. Chapman, G. Jost, and R. Pas van der. Using OpenMP: Portable Shared Memory Parallel Programming. Cambridge, MA, USA: MIT Press, 2007.

[42] M. Costantini, "A novel phase unwrapping method based on network programming," IEEE Trans. Geosci. Remote Sens., vol. 36, no. 3, pp. 813–821, May 1998.

[43] J. Gustafson, "Reevaluating Amdahl's law," Commun. ACM, vol. 31, pp. 532–533, May 1988.

[44] S. Elefante et al., "SBAS-DINSAR time series generation on cloud com- puting platforms," in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS'13), Melbourne (AU), Jul. 2013, pp. 274–277.

[45] Zebker, H.A.; Villasenor, J. Decorrelation in interferometric radar echoes. IEEE Trans. Geosci. Remote Sens. 1992, 30, 950–959.

Chapter 3

A First Assessment of the P-SBAS DInSAR Algorithm Performances Within a Cloud Computing Environment

We present in this chapter a first performance assessment of the Parallel Small BAseline Subset (P-SBAS) algorithm, for the generation of Differential Synthetic Aperture Radar (SAR) Interferometry (DInSAR) deformation maps and time series, which has been migrated to a Cloud Computing (CC) environment. In particular, we investigate the scalable performances of the P-SBAS algorithm by processing a selected ENVISAT ASAR image time series, which we use as a benchmark, and by exploiting the Amazon Web Services (AWS) CC platform. The presented analysis shows a very good match between the theoretical and experimental P-SBAS performances achieved within the CC environment. Moreover, the obtained results demonstrate that the implemented P-SBAS Cloud migration is able to process ENVISAT SAR image time series in short times (less than 7 h) and at low costs. The P-SBAS Cloud scalable performances are also compared to those achieved by exploiting an in-house High Performance Computing (HPC) cluster, showing that nearly no overhead is introduced by the presented Cloud solution. As a further outcome, the performed analysis allows us to identify the major bottlenecks that can hamper the P-SBAS performances within a CC environment, in the perspective of processing very huge SAR data flows such as those coming from the existing COSMO-SkyMed or the upcoming SENTINEL-1 constellation. This chapter describes a relevant step toward the challenging Earth Observation scenario focused on the joint exploitation of advanced DInSAR techniques and CC environments for the massive processing of big SAR Data.

3.1 The Current DInSAR Scenario

Advanced differential synthetic aperture radar (SAR) interferometry (DInSAR) usually identifies a set of algorithms, tools, and methodologies for the generation of Earth's surface deformation maps and time series computed from a sequence of multitemporal differential SAR interferograms [1]. A widely used advanced DInSAR approach is the technique named Small BAseline Subset (SBAS) [2] which allows to produce line-of-sight (LOS) projected mean deformation velocity maps and corresponding displacement time series by exploiting interferograms characterized by a small temporal and/or spatial separation (baseline) between the acquisition orbits. The SBAS algorithm has proven its effectiveness to detect ground displacements with millimeters accuracy [3] in different scenarios, such as volcanoes, tectonics, landslides, anthropogenic induced land motions [4]–[7] and it is capable to perform analyses at different spatial scales [8] and with multisensor data [9], [10]. The SBAS algorithm, and more generally the advanced DInSAR techniques, found their success on the large avail- ability of SAR data archives acquired over time by several satellite systems. Indeed, the current radar Earth Observation (EO) scenario takes advantage of the widely diffused long- term C-band ESA (e.g., ERS-1, ERS-2, and ENVISAT) and Canadian (RADARSAT-1/2) SAR data archives, which have been acquired during the last 20 years, as well as of data sequences provided by the X-band generation SAR sensors, such as the COSMO-SkyMed (CSK) and TerraSAR-X (TSX) constellations. Moreover, a massive and ever increasing data flow will be further supplied by the recently launched (April 2014) Copernicus (European Union) SENTINEL-1A SAR satellite, which will also be paired during 2016 with the SENTINEL-1B twin system that will allow halving the constellation revisit time (from 12 to 6 days) [11]. With the SENTINEL-1 era, new SAR data relevant to extended areas on Earth will be soon publically available, thanks to the free and open access data policy adopted by the Copernicus program.

Moreover, the SENTINEL-1 data will be collected on land by using the TOPS SAR mode that has been specifically designed for DInSAR and advanced DInSAR applications [12], [13].

In this context, the massive exploitation of these Big SAR data archives for the generation of advanced DInSAR products will open new research perspectives to understand Earth's surface deformation dynamics at global scale.

However, the accomplishment of this task requires not only large and very high computing resources to process the existing and upcoming huge SAR data amounts within short time frames, but also efficient algorithms able to effectively exploit the available computing facilities.

To provide a contribution toward this direction, a parallel version of the SBAS algorithm, namely parallel SBAS (P-SBAS), has been proposed [14]. P-SBAS that is extensively discussed in Chapter 2, permits to generate, in an automatic and unsupervised manner, advanced DInSAR products by taking full benefit from parallel computing architectures, such as cluster and GRID infrastructures. P-SBAS has been extensively tested by exploiting in-house processing facilities achieving promising results in terms of both scalability and efficiency [14]–[16]. However, it is worth noting that, even if in-house solutions can provide high computing performances, they can represent a bottleneck due to their intrinsic limited resource availability. Therefore, they cannot be suited to properly face the massive processing that will be inevitably required by the expected huge SAR data flow, particularly when global scale analyses are concerned. Moreover, in-house High Performance Computing (HPC) infrastructures can be very expensive in terms of procurement, maintenance, and upgrading.

The use of Cloud Computing (CC) environments represents a promising solution to overcome the above-mentioned limitations and this is the rationale why they are becoming more and more diffused in EO scenarios [17]–[19]. Indeed, CC provides highly scalable and flexible architectures that are, in general, computationally very efficient and less expensive with respect to in-house solutions. An overview about the main characteristics of cloud environment is sintetically decribed in Appendix-A.

In addition, CC can be extremely helpful for both resources optimization and performance improvements due to its possibility to build up customized computing infrastructures. Moreover, the increasing availability of public Cloud environments [20]-[22], and their relative simplicity to use, thanks to advanced application programming interfaces (API) and web-based tools, is further pushing toward the use of such a technology also in scientific applications [19], [23], [24]. In this context, the migration of scientific applications to CC environments is, therefore, a key issue, with particular reference to advanced DInSAR algorithms, because a significant initial effort and a deep preliminary analysis of the specific algorithm which has to be "cloudified" could be required.

We present in this chapter a first performance assessment of the P-SBAS algorithm that has been migrated to a public CC environment. In particular, the goal of our study is to evaluate the P-SBAS scalable performances achieved within a CC environment as well as to identify the major inefficiency sources that can hamper such performances. To this aim, we select the proper Cloud migration approach that allows the full exploitation of the P-SBAS parallelization strategy. Moreover, due to the high P-SBAS computational requirements, we evaluate the most appropriate Cloud resources configuration, in terms of both instances and storage. An extensive analysis is carried out by processing a selected SAR images time series acquired (from ascending orbits) by the ENVISAT ASAR sensor over the Napoli bay area and by exploiting the Amazon Web Services (AWS) Cloud platform. The obtained P-SBAS scalable performances are also compared to those achieved by exploiting an in-house, dedicated, HPC cluster.

In addition an estimation of the costs relevant to the experimental test performed on the AWS cloud is carried out in order to evaluate the trade-off between processing times and costs of an interferometric computation depending on the employed resources.

3.2 P-SBAS processing chain description

We provide in this section a concise description of the P-SBAS processing chain aimed at presenting its main processing steps. In particular, each step is briefly described in terms of main tasks, implemented procedures and computational issues relevant to CPU usage, RAM occupation and I/O transfer requirements.

The P-SBAS solution has been designed by carefully taking into account several aspects, such as data dependencies, task partitioning, inherent granularity, scheduling policy, load unbalancing and I/O. It is worth emphasizing that, due to the heterogeneous nature of the algorithms comprised within SBAS processing chain, appropriate parallelization strategies have been adopted depending on the algorithmic structure of the particular processing step under evaluation [14]

In order to take advantage from both multi-node and multi-core architectures, two-parallelization levels have been considered, i.e., the process and the thread level. The former considers a coarse/medium granularity based approach (mainly applied to the whole processing chain) while the latter relies on a fine-grained parallelization and it has been implemented for the most computing-intensive processing operations [14].

The block diagram of the P-SBAS processing chain is shown in Figure 3.1; note that in this scheme the steps depicted by red blocks represent the jobs that are parallel executed by simultaneously running on different computing units. Moreover, dashed line blocks depict the steps which also exploit multi-core parallelism making use of multithreading programming techniques. Finally, the black blocks represent steps that are intrinsically sequential as they merge together information and data coming out from the previous steps executions. For each step of Figure 3.1, the main characteristics in terms of CPU and RAM usage and I/O operations are also summarized in Table 3.1.

Let us start our discussion by considering the SAR data focusing operation implemented by the step A

that consists in transforming the radar raw data into microwave images, often referred to as Single Look Complex (SLC) images [25]. This step has a high computational burden because it mostly performs two-dimensional FFTs of large complex data matrices on the order of GBytes [15]; moreover, it also has a remarkably significant data flow as it involves seldom I/O operations (see Table 3.1).

Step B performs the Digital Elevation Model (DEM) conversion into SAR coordinates that consists in referring the elevation profile of the area under investigation with respect to the radar imaging grid. In this step, data matrices typically of the order of several GBytes are processed and, therefore, this step requires a remarkable amount of memory and I/O operations.

Within step C, the SAR image co-registration operation is carried out to refer all the data to a unique radar geometry via an interpolation of all the focused images with respect to a reference one referred as "master" image [26]. Also for this step, the main limitation is represented by the file dimension, which requires intensive (hundred of GBytes) I/O access and significant (nearly tens of GBytes) RAM memory (see Table 3.1).

The step D performs the identification of the interferometric data pairs required for the subsequent co-registration refinement step E, in which possible residual sub-pixel shifts of the images are firstly evaluated and, afterwards, used to resample the images. This step is particularly I/O demanding (hundreds of GBytes to read/write) due to the applied resampling method based on FFTs computed on full resolution SAR images.

Before moving to the interferogram generation, the evaluation of some parameters useful for the subsequent steps is carried out (block F of Figure. 3.1). Having both the co-registered images (steps C and E) and the DEM (as output of step B) referenced to a common radar geometry, the differential interferograms are generated through step G.

These operations require a high memory RAM usage (tens of Gbytes) as well as intensive I/O workload (hundreds of GBytes) as they are carried out at the SAR images' full spatial resolution and in

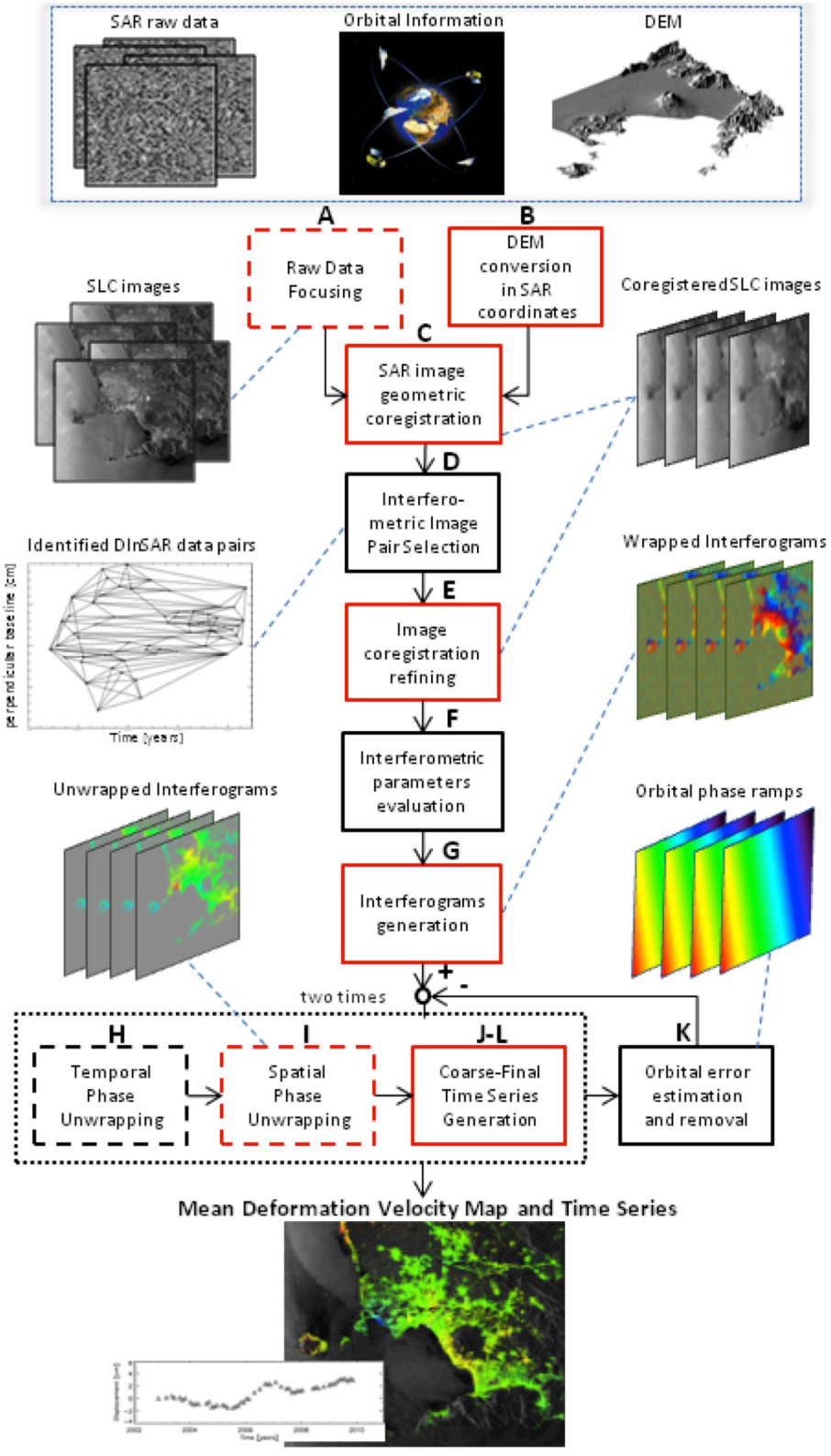


Figure 3.1: P-SBAS workflow. Black and red blocks represent sequential and parallel (from a process-level perspective) processing steps, respectively. Dashed line blocks represent multi-threading programmed processing steps.

the complex domain. the complex domain. However, the final results are stored in low resolution

mode following a complex spatial average (multi-look) operation [15]; this procedure mitigates the noise (decorrelation) affecting the DInSAR interferograms and drastically reduces the sizes of the final outputs, but not those of the intermediate products.

The modulo-2π restricted phase of each computed multi-look interferogram needs to be "unwrapped" to retrieve the original phase [26]. This procedure is carried out in step H and I by applying the extended minimum cost flow (EMCF) phase unwrapping (PhU) algorithm [27].

The phase unwrapping steps are the most demanding in terms of memory (tens of GBytes of RAM required) and CPU usage (it exploits multithreading programming) (see Table 3.1). Indeed, they deal with wrapped and unwrapped interferogram stacks represented by three-dimensional (3-D) matrices.

A pixel-based inversion of the unwrapped phases of the interferogram stack is, afterwards, carried out (step J) to retrieve the deformation time-series. Moreover, in step K, the estimation of possible residual phase artifacts affecting the interferograms, often referred to as "orbital phase patterns", is undertaken. Such phase ramps that are due to possible orbital inaccuracy, are subsequently removed from the wrapped interferograms and a final PhU step has to be performed on the “orbital error free” interferograms (second run of step H, I, and J of Figure 3.1). Block L provides the final deformation time-series after executing the temporal coherence estimation [27], which provides information about the quality of the retrieved displacement measurements. In the rest of this section we summarize the most critical issues concerning the porting of the parallel SBAS algorithm within distributed computing environments. First of all, the P-SBAS approach requires the execution of different steps to be performed in a specific and fixed order. For instance, SAR image registration can occur only after the raw data focusing operation has been accomplished, or the unwrapping step can be executed only after the generation of the differential interferograms sequence.

This rationale shows that the successively generated data and products of the SBAS processing chain are strongly interconnected.

Table 3.1 P-SBAS workflow requirements for a standard SAR data set relevant to the Napoli Bay area (64 ENVISAT images)

Step	CPU usage	Maximum Resident Set Size (MB)	I/O Workload (MB)
A. Raw Data Focusing	53%	3700	928000
B. DEM Conversion	37,5%	9400	83000
C. SAR Image registration	14%	7800	592000
D. Interferometric Pair selection	12,5%	280	2100
E. Image coregistration refining	22,5%	5600	226000
F. Interferometric parameters evaluation	34%	6000	2500
G. Interferograms generation	25%	13000	686000
H. Temporal phase unwrapping	72,5%	9000	53000
I. Spatial phase unwrapping	72,5%	22000	80000
J. Deformation and residual topography estimation	12,5%	5800	2160
K. Orbital error estimation	19%	7200	6430
L. Displacement time series	15%	6000	12600

These values refer to a standard data set (64 ENVISAT images, see Section 4.1).The CPU usage percentage values refer to the percentage of CPU that each step got on average per single node. Note that the represented values have been normalized with respect to the maximum achievable CPU usage, which in our case is given by the full exploitation of 8 cores per node. The I/O workload is calculated as the total of data read and written per step and is given in MegaBytes.

Moreover, in some steps the joint processing of the outputs generated by previous blocks need to be performed. Hence, a strategy that ensures the sharing of the middle chain products among the different steps is needed. Secondly, many steps of the processing chain, such as the DEM conversion (step B), the interferogram generation (step G) and the phase unwrapping steps (H and I), are characterized by large RAM usage, as detailed in Table 3.1. This issue places specific requirements on the computing instances that have to be provisioned to execute these steps. Finally, very large amounts of data are involved in the SBAS processing - hundreds of GBytes for the considered standard dataset which can become TBytes if COSMO-SkyMed or SENTINEL-1 datasets are concerned - and, therefore, data transfer and I/O issues are two critical aspects to be carefully taken into account. Hence, dedicated storage capability and network facility need to be exploited. The P-SBAS processing chain has been, until now, only extensively tested on a dedicated cluster that is located at the CNR-IREA premises where it has originally been developed. The P-SBAS technique has already been broadly used to process the ENVISAT SAR data archives1 but its rationale is also well suited to process the upcoming SENTINEL-1 data stream.

3.3 Cloud Migration

As known in the literature, there are different possibilities for migrating an existing application in a cloud-computing environment that comprise either adaptation or re-design of the application components. More specifically, according to [28], the following four major approaches can be identified:

i. Replace components with Cloud offerings;

ii. Partially migrate some of the application functionality to the Cloud;

iii. Migrate the whole software stack of the application to the Cloud;

iv. Cloudify the application.

Obviously, each of the above-mentioned solutions provides advantages and dis-advantages and the choice is driven by the evaluation of the tradeoff among different aspects, including the migration effort, the infrastructure costs, and the overall performance (not only the total computation time). In our case the goal was to evaluate the performances that are achieved within a Cloud environment, by directly comparing them to those obtained on a HPC cluster. Moreover, we aimed to investigate the costs associated with the Cloud exploitation for the P-SBAS processing in order to evaluate its convenience. Therefore, at this early stage, we opted for a migration approach that allows us to straightforwardly attain such aims by minimizing the deployment effort at the same time. Before managing the actual migration of the P-SBAS application to cloud, a deep analysis of the workflow that takes into account both software and hardware requirements was carried out. The P-SBAS algorithm has a complex workflow that is composed of several steps both sequential and parallel (see Figure 1). The majority of the P-SBAS chain algorithms are developed in the Exelis Interactive Data Language (IDL) [29]: a scientific programming language that combines numerical and graphical abilities and is widely used by engineers and scientists developing algorithms for SAR and DInSAR data processing. IDL is a commercial software and therefore each machine running the application requires a license. As extensively described in chapter 2, the P-SBAS rationale provides that in many steps of the chain the joint processing of the outputs generated by previous steps needs to be performed; hence, a common storage and an efficient management of data sharing between the processing nodes are required. For this purpose, Network File System (NFS) was adopted, which is a distributed file system protocol that allows exploiting a network to access to a remote storage for saving and updating files, as though they were on the local storage [30].

According to the specific P-SBAS requirements, we limited the possible migration choices to the last two approaches (iii and iv) and evaluated their respective benefits and drawbacks. The iii) approach is

the most common example of migration to the Cloud, where the application is easily encapsulated in virtual machines (VMs) that run on the Cloud. Such an approach implies the minimum migration effort and allows us to easily evaluate the benefits, in terms of processing times and costs, to exploit the Cloud for scientific applications. Indeed, the Cloud adoption could be a valid alternative with respect to the need to update/buy new clusters, which are very expensive, require maintenance and rather quickly become obsolete.

Furthermore, thanks to the use of VM, several benefits are gained, which are in order:

1. Security and Isolation: it runs services in cloud environment totally independent from each other;
2. Ease of Administration and Management: due to the common virtualization layer and the adoption of snapshots for both installation and configuration;
3. Disaster Recovery: VM can be launched in a few minutes and furthermore, can be cloned and migrated to different locations;
4. High Reliability and Load Balancing Optimization.

The iv) approach could be interesting as, in addition to the previous ones, some benefits would be expected from the exploitation of software using specific cloud features (as for example Hadoop or Cloud orchestration software). However, although the adoption of the MapReduce-based framework seems to be promising in managing large clusters of data, results from the literature [31], [32] show that this is particularly true when it is possible to pre-process data to optimize both the mapping and reducing phases. Indeed, the mapping phase may require a number of steps that could be automatized through a workflow of jobs (e.g Hadoop workflow) but the data constraints and the scheduling policies should be properly tuned. In particular, concerning the interferometric processing, the data mapping phase is easy because of two main constraints. First of all it is not always possible to split the input data of the P-SBAS processing steps in smaller portions to be distributed to all available nodes because

of the spatial dependancies. Secondly, as already mentioned, some intermediate steps of the chain require access to all data products produced by the previous steps. Based on the above considerations, we estimated that the cloudification effort should require a separate analysis to better gain benefit from the cloud and this will be addressed in future work. For all these reasons, the approach of migrating the whole P-SBAS software stack to the Cloud was adopted to port the P-SBAS processing chain to the Cloud environment.

3.4 Cloud Deployment

As a cloud-computing infrastructure for the P-SBAS algorithm deployment we chose AWS Elastic Compute Cloud (EC2) because it is currently a feature-rich, stable and commercial public cloud [20]. AWS is a web service that allows users to easily configure and instantiate VM images.

Moreover, AWS EC2 is based on the XEN para-virtualization technology [33], which also provides the capability of sizing instances based on the EC2 Compute Unit (ECU).

We implemented an Amazon Virtual Private Cloud (VPC) that is a logically isolated section of the AWS in which resources can be launched in a completely defined virtual network. The easy customization of the network configuration allows the users to fully control the virtual networking environment through IP address range selection, subnet creation, route tables configuration, network gateways and multiple layers of security (security groups and network access control lists). Indeed, thanks to this customization, server linking and SAR data uploading have been configured with predefined rules to ensure a secure connection between end points and to allow only authorized users to share data.

Moreover, we configured a customized Amazon Machine Image (AMI) containing the operating system (Ubuntu distribution), the packages enabling an easy integration with AWS services (i.e.

Amazon EC2 API and AMI tools), and the software needed for the P-SBAS processing, such as IDL. Note that this software can run with a license that is available on a local server (IDL licenses server) and we need that all AMIs in the Cloud securely communicate with this server to gather the proper license.

In order to automate as much as possible the provisioning and configuration of cloud resources, (e.g. automatically handle EC2 instances, network and storage facilities), we developed Linux Bash scripts implementing the AWS Command Line Interface (CLI) which is a unified tool provided by AWS to manage multiple services [20]. The goal of these scripts is twofold; on the one hand, we need to set up the configuration to run the P-SBAS algorithm, on the other hand, we want to connect, start and stop additional computing nodes (workers) in order to perform the parallel tasks of the algorithm on different dedicated machines.

More specifically, through the aforementioned Linux Bash scripts, one of the instances is configured as the master node and performs multiple tasks. In particular it handles the whole P-SBAS workflow by managing the scheduling of parallel jobs that will be executed by the worker nodes, it performs computation, and it acts as an NFS server (see Figure 5.2). The implemented scripts, also allow the selection of the number and type of EC2 instances to provision. A dynamical use of the employed EC2 instances is accomplished by allowing to start/stop them according to the parallel/sequential parts of the algorithm and to terminate all the instances at the end of the processing.

This solution allows us to optimize the EC2 instances exploitation, and therefore the associated costs, because they are maintained active only when operative, thus avoiding that they remain idling while other nodes are working. Also, the worker nodes configuration is automatically performed by connecting each of them, as soon as the instance is started, both to the master node through an NFS client as well as to the IDL licence server.

Moreover, the implemented Bash script automatically performs the creation of the selected storage

volume. Such an external disk is specifically configured to support the P-SBAS I/O workload and it is mounted to the master node.

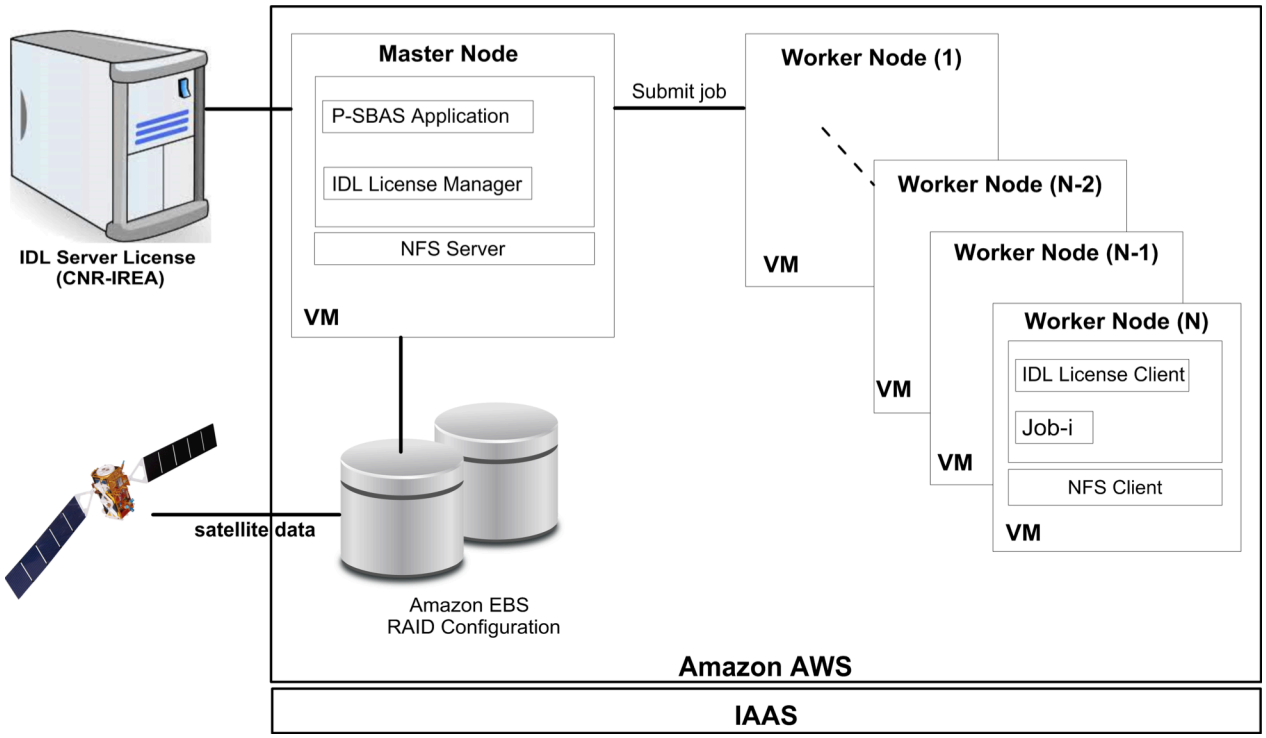


Figure 3.2: Public Cloud Platform for DInSAR analysis; it consists of one master and N worker nodes as well as a common storage volume that contains the input data. All the components are located in the Amazon cloud.

3.5 Experimental analysis

In this section we present an experimental analysis aimed at investigating the P-SBAS processing performances that are achieved by using two different cloud resources configurations within the AWS environment (in terms of instance type and storage) when the number of exploited instances increases. The goal of our study is twofold; on the one hand, we evaluated the cloud processing elapsed times by also comparing them with the corresponding processing times obtained on a HPC cluster. On the other

hand, we carried out an estimation of the costs relevant to the experimental tests that have been performed on the AWS cloud in order to evaluate the trade-off between elapsed times and costs of an interferometric SAR data processing, depending on the exploited resources. Such an analysis could be very useful to select the best configuration in terms of typology and amount of cloud resources to use according to the specific time/cost requirements.

3.5.1 Dataset and P-SBAS results

The experimental analysis has been carried out by exploiting an interferometric SAR data stack acquired over the Napoli bay area, a densely urbanized zone located in Southern Italy and including the active caldera of Campi Flegrei, the Vesuvius volcano, and the city of Napoli. In particular, 64 ENVISAT SAR data acquired from ascending orbits, spanning the 2002–2010 time interval and approximately covering an area of 100x100 km^2 (corresponding to an ENVISAT frame), have been considered. The overall input data, comprising the SAR raw data sequence; the DEM files and the orbital information come to approximately 20 GBytes. In Figure 3.3 we show the mean deformation velocity map in the radar line of sight (LOS), which is one of the DInSAR results generated through the P-SBAS processing. It is represented in the radar slant-range coordinates and afterwards superimposed on a SAR amplitude image of the investigated area. The estimated mean deformation velocity has been only computed in the areas in which the measurement accuracy is not affected by decorrelation noise (coherent pixels). In particular, it is worth noting that in Figure 3.3(a) a significant deformation pattern corresponding to the area of the Campi Flegrei caldera is clearly shown. Moreover, the computation of the temporal evolution of the detected deformation has also been carried out for each coherent pixel of the scene. For instance, the chronological sequence of the computed displacement of a specific pixel (located in the maximum deforming area of the Campi Flegrei caldera) is plotted in Figure 3.3(b).

These results are in accordance with ground truth measurements [9], [10]. The maximum storage capacity required by this dataset processing is of about 450 GBytes.

3.5.2 P-SBAS performance analysis within AWS cloud

The Napoli Bay area dataset has been chosen because it was already used as a benchmark for the evaluation of the P-SBAS parallel performances on a HPC cluster [14]. Hereafter, we recall the major results that were obtained in order to introduce the analysis presented in this paper, which has been carried out on the AWS cloud. In [14] we thoroughly studied the scalable performances of the overall P-SBAS algorithm by evaluating its speedup with respect to the number of computing elements.

As a computational platform, the HPC cluster that is located at the CNR-IREA premises was used. From this analysis it turned out that the major source of inefficiency for the P-SBAS parallel performances was ascribable to the presence of a non-negligible intrinsic processing sequential fraction, which, even if it is liable to a further reduction, remains essentially non-eliminable because of the complex nature of the P-SBAS algorithm. Apart from that factor which was quantitatively taken into account by computing the relevant P-SBAS Amdahl's law [34], [35], the speedups that were achieved, by exploiting from 1 up to 16 computing nodes, showed to be definitely satisfactory (with a maximum deviation from the Amdahl's law of about 18% in correspondence with 16 nodes).

The CNR-IREA cluster used for the scalability analysis consists of 16 nodes, each equipped with 2 CPUs (eight-core 2.6 GHz Intel Xeon E5-2670) and 384 GB of RAM (see Table 3.2). The cluster has a storage shared among different nodes that is implemented through NFS and employs a 56Gbit/s InfiniBand interconnection. In particular, each processing node is equipped with a Direct Attached Storage (DAS) system in a RAID 5 configuration that ensures a disk access bandwidth of approximately 300 MB/s. The aim of this experimental section is to investigate the P-SBAS

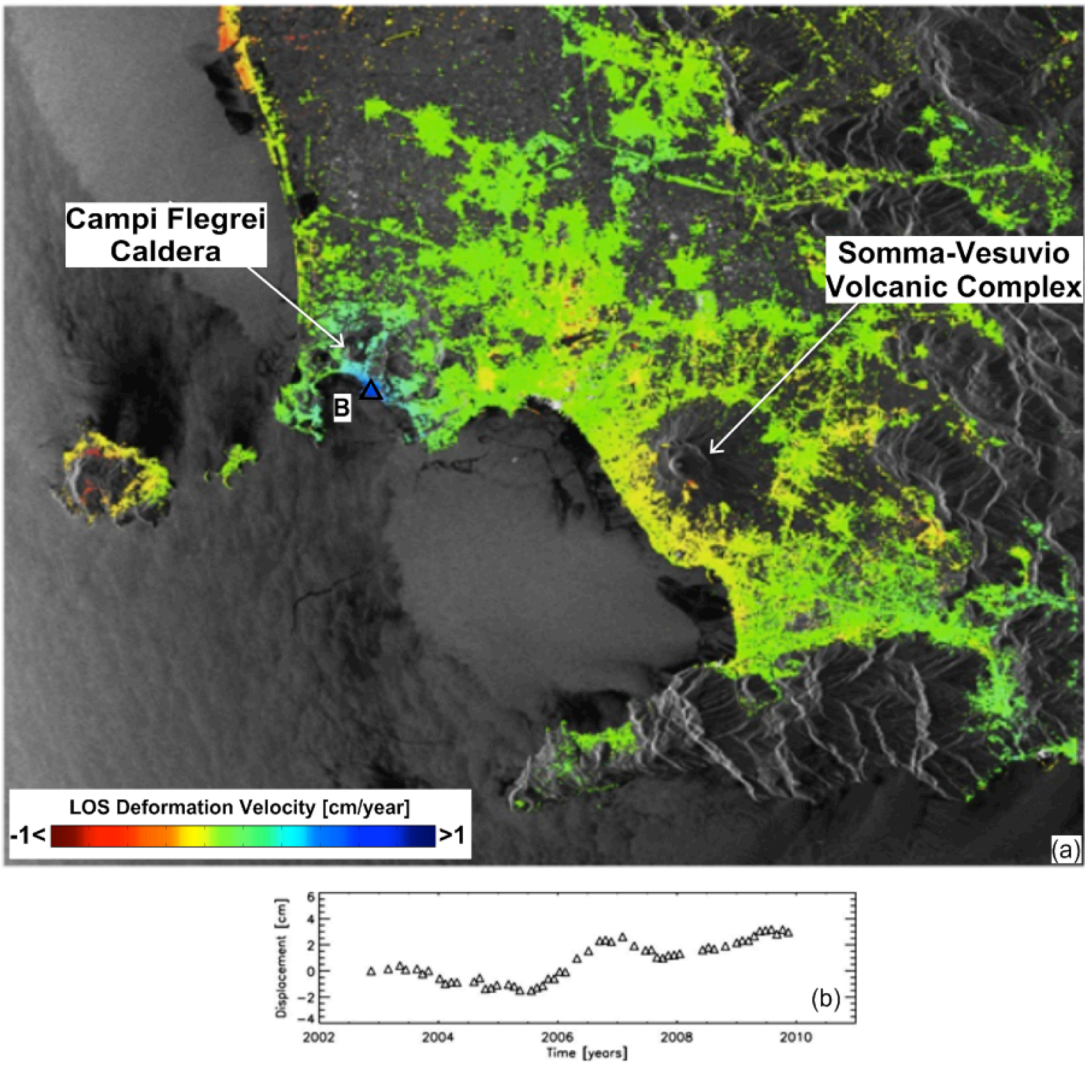


Figure 3.3: (a) Mean deformation velocity map of the Napoli Bay area, in line of sight (LOS), which has been generated via the P-SBAS processing on Amazon Cloud. (b) Plot of the retrieved displacement time series relevant to a pixel located in the area of maximum deformation of the Campi Flegrei Caldera.

performances,in terms of processing times, when the number of computing nodes increases, within the AWS cloud by using different resource configurations and comparing them to the corresponding performances achieved on the CNR-IREA cluster.

We carried out such an evaluation by exploiting, among the available EC2 instances, those which have similar characteristics to the used cluster nodes and which satisfy the P-SBAS computing requirements to process the selected data-set. Note that, from here on, we refer to an EC2 instance as to a single computing node.

Table 3.2
CNR-IREA cluster nodes and Amazon Instance Type Characteristics

	CNR-IREA cluster nodes	m2.4.xlarge	c3.8.xlarge
Processor	Intel Xeon E5-2670	Intel Xeon X5550	Intel Xeon E5-2680
cores	8 (used)	8	8 used
RAM	384 GB	68 GB	60 GB
Network	Infiniband (56 Gb/s)	1 Gb/s	10 Gb/s
NFS storage bandwidth	300 MB/s	128 MB/s	256 MB/s

* Note that although the cluster nodes are equipped with 16 cores and the c3.8xlarge instances with 32 cores, only 8 have been used for the experimental tests to achieve a proper comparison between the different platforms.

Table 3.3
P-SBAS processing elapsed times on private cluster and on AWS cloud relevant to the Napoli Bay area dataset

Nodes	CNR Cluster processing time (hours)	m2.4xlarge processing time (hours)	c3.8xlarge processing time (hours)
1	41	54	40.5
2	23	31	22
4	14	18.5	13.5
8	9	12.5	8.5
16	7	-	6.5

In accordance with our objective, we took both the m2.4.xlarge and c3.8.xlarge instances into consideration because they have both processors that are comparable to the cluster nodes ones, as well as enough RAM to run P-SBAS without incurring *page-faulting*; they are, however, characterized by different network bandwidths (see Table 3.2).

Actually, the network performance is a key factor to guarantee a good scalability; indeed, one of the major critical issues of the P-SBAS algorithm is the very large amount of data (in terms of inputs, intermediate products and final results) that are read/written during the overall processing into the common NFS storage. This implies that, when the number of exploited instances increases, the number

of concurrent processes, which simultaneously read or write on the common storage volume, increases as well, thus remarkably raising the I/O workload.

In this case, the network bandwidth can become a bottleneck. Furthermore, for the same reason, we carefully evaluated the Amazon Elastic Block Store (EBS) volume type to be used as a common NFS storage. More specifically, as a first test bed we exploited the m2.4xlarge instances for both master and worker nodes and we used, as a storage disk, one "provisioned IOPS" (SSD) EBS volume which guarantees 4000 IOPS (I/O operation per second), i.e. a disk access bandwidth of about 128 MB/s (see Table 3.2). In Table 3 the elapsed times relevant to the P-SBAS processing carried out by exploiting from 1 up to 8 nodes with this configuration are presented, together with the corresponding times related to the P-SBAS runs, which have been performed on the CNR-IREA cluster.

The experimental results achieved on the cloud with the m2.4xlarge instances showed computing performances worse than those obtained on the cluster. This is ascribable, in a small part, to the m2.4xlarge processor that is slightly worse than the cluster nodes one, but even more so it is ascribable to the bottleneck due to the network/disk bandwidth which already begins saturating at 4 nodes in correspondence to the P-SBAS steps that are heavier in terms of I/O workload and becomes fully saturated at 8 nodes. Hence, it does not make sense to carry out the test with 16 nodes by exploiting this configuration.

In order to overcome the above mentioned limitation, as a second test bed we used the c3.8xlarge instances which have a 10Gb/s network bandwidth and, as a storage, two "provisioned IOPS" (SSD) EBS volumes in a RAID 0 configuration which allows achieving a total throughput of 8000 IOPS (I/O operation per second) that translates in a disk access bandwidth of about 256 MB/s [20] (see Table 3.2). The fourth column of Table 3.3 represents the elapsed times relevant to the P-SBAS processing carried out by exploiting from 1 up to 16 nodes with the c3.8xlarge configuration. It is evident that in this case the performances that are achieved on cloud are definitely comparable to those obtained on the CNR-

IREA cluster. Hence, we can conclude that by exploiting the proposed configuration, the cloud environment does not introduce further overhead to the P-SBAS parallel implementation, thus allowing us to take full advantage of P-SBAS scalable performances.

The histograms in Figure 3.4 and 3.5 show in detail the computing times for each step of the P-SBAS processing chain relevant to the runs with 8 and 16 nodes respectively. In particular, in Figure 3.4 the performances achieved by exploiting the cluster, the m2.4xlarge and the c3.8xlarge configurations are compared, while in Figure 3.5 those relevant to the cluster and the c3.8xlarge are juxtaposed. It is evident from Figure 3.4 that the m2.4xlarge configuration always presents the worst performances, but it is worth noting that the largest elapsed times are found in correspondence with the most I/O demanding P-SBAS processing steps, i.e. A, C and G steps (see Figure 3.1).

Actually, these are also the only steps in which the CNR-IREA cluster performances are better than the c3.8xlarge configuration ones, due to the cluster's larger network/disk bandwidth. This behavior is accentuated for the 16 nodes run depicted in Figure 3.5 where the I/O workload is doubled, with respect to the 8 nodes run, in correspondence with the P-SBAS parallel steps. In this case, at the A, C, and G steps of the P-SBAS processing chain, the disk access bandwidth of the RAID storage volume becomes saturated. The obtained results suggest that, concerning P-SBAS elaborations of data-sets similar to the one that we are considering, exploiting a greater number of nodes would not be of use; while, in the perspective of processing larger datasets which require parallelization on more nodes, a storage volume with a higher I/O throughput can be used by a RAID configuring more than two EBS volumes (up to now, it is possible to implement a RAID 0 configuration with up to 8 volumes).

Actually, these are also the only steps in which the CNR-IREA cluster performances are better than the c3.8xlarge configuration ones, due to the cluster's larger network/disk bandwidth. This behavior is accentuated for the 16 nodes run depicted in Figure 3.5 where the I/O workload is doubled, with respect to the 8 nodes run, in correspondence with the P-SBAS parallel steps.

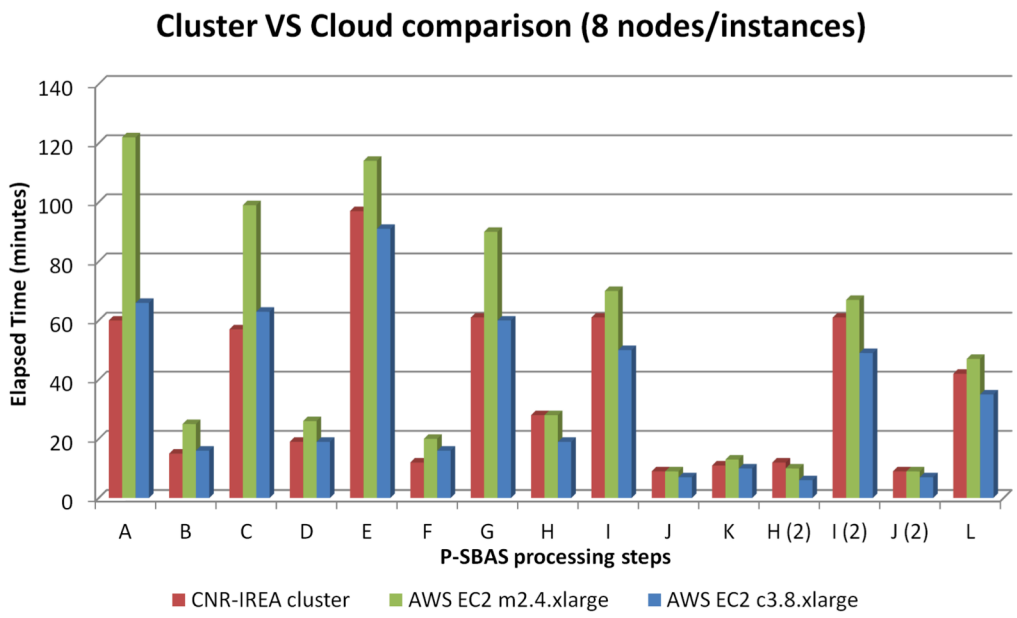


Figure.3.4: P-SBAS steps processing times relevant to the CNR-IREA cluster run (red bars), the AWS-m2.4xlarge run (green bars) and the AWS-c3.8xlargerun (blue bars) by exploiting 8 nodes/instances.

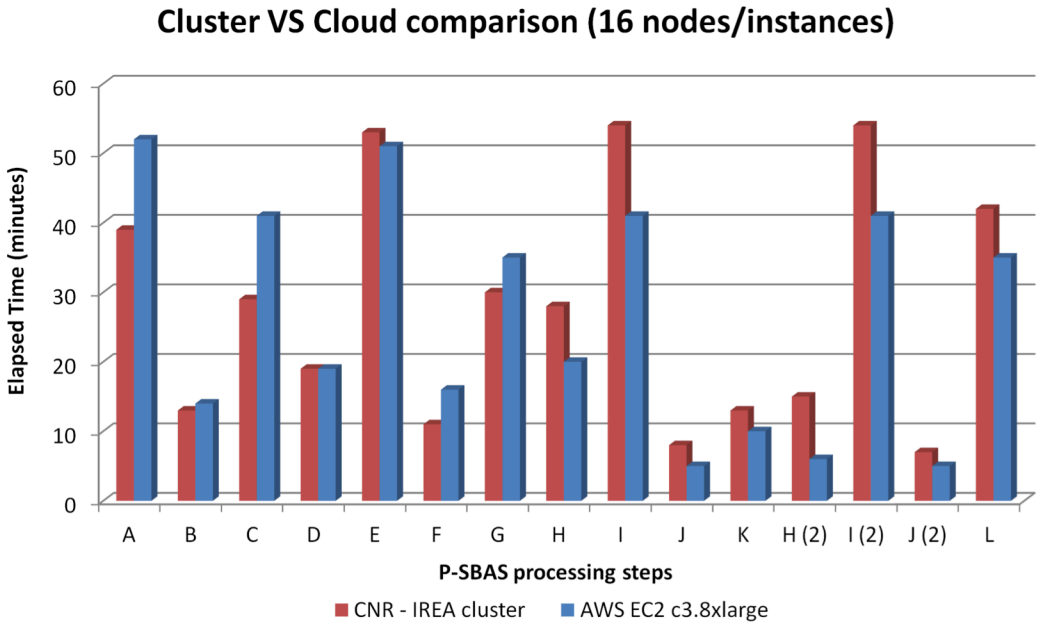


Figure 3.5: P-SBAS steps processing times relevant to CNR-IREA cluster run (blue bars) and the AWS-c3.8xlarge run (red bars) by exploiting 16 nodes/instances.

In this case, at the A, C, and G steps of the P-SBAS processing chain, the disk access bandwidth of the RAID storage volume becomes saturated. The obtained results suggest that, concerning P-SBAS elaborations of data-sets similar to the one that we are considering, exploiting a greater number of nodes would not be of use; while, in the perspective of processing larger datasets which require parallelization on more nodes, a storage volume with a higher I/O throughput can be used by a RAID configuring more than two EBS volumes (up to now, it is possible to implement a RAID 0 configuration with up to 8 volumes). On the contrary, concerning the most CPU demanding steps of P-SBAS, i.e. I and H, Figure 3.1 clearly shows that the times elapsed by exploiting the CNR-IREA cluster are greater than those obtained with the c3.8xlarge configuration. This is due to the slightly better processor of c3.8xlarge instances with respect to the one with the cluster nodes (see Table 3.2).

3.5.3 Speedup Analysis

A scalability analysis with respect to the number of exploited computing nodes has been carried out both on the CNR-IREA cluster and on AWS Cloud. Concerning the test performed at the CNR-IREA premises, the speedup depicted as a function of the number of engaged nodes is represented in Fig. 4. Such a plot shows the speedup ideal linear behavior (blue/diamonds), the Amdahl's law (red/squares) and the experimental results achieved on the above-mentioned cluster (green/triangles). The Amdahl's law has been evaluated by computing the processing sequential fraction as the ratio between the sum of elapsed times of the P-SBAS sequential parts and the total processing time on a single computing node; it has turned out to be approximately 9% ($f_s = 0.09$). It is evident from Figure 3.6 that the achieved speedup is definitely satisfactory; indeed the speedup curve is very close to the Amdahl's law with a slight deviation as approaching 16 nodes. This result reveals that, by exploiting a HPC cluster, all the factors that can hamper the efficiency, such as load unbalancing, communication times and I/O

Table 3.4
Values of the Amdahl's law and experimental speedup on both CNR-IREA cluster and AWS cloud; their percentage deviation is also shown.

Number of Nodes	1	2	4	8	16
CNR-IREA cluster Amdahl's law	1	1.84	3.15	4.92	6.84
CNR-IREA cluster actual Speedup	1	1.80	2.90	4.26	5.60
AWS cloud Amdahl's law	1	1.86	3.27	5.25	7.55
AWS cloud actual Speedup	1	1.83	3.07	4.72	6.20
Deviation between Amdahl's law and actual Speedup (%) CNR-IREA	0	1.97	7.90	13.46	18.23
Deviation between Amdahl's law and actual Speedup (%) AWS	0	1.67	6.16	10.19	17.82

overhead [36], are basically negligible at least up to 16 nodes. Hence the only significant inefficiency source is the P-SBAS sequential fraction that is taken into account by the Amdahl's law and is still subject of further improvements, being some sequential parts of the P-SBAS processing chain currently on progress to be turned into parallel ones.

Figure 3.7, instead, shows the speedup that has been evaluated through the scalable analysis carried out within AWS Cloud by exploiting up to 16 instances. In this case, the processing inherent sequential fraction has been estimated to be approximately 7,5% ($f_s = 0.075$), thus being even slightly smaller than in the cluster case.

Such a difference is ascribable to the fact that the processing elapsed times strongly depend on the exploited specific computational environments that in the cluster and Cloud cases are different; in particular, the processors of the Cloud nodes are slightly more powerful than those of the cluster nodes (see Tables 3.2). As Figure 5.5 clearly shows, also in this case, the speedup behavior is very close to the Amdahl's law and it begins to diverge as approaching 16 nodes. In Table 3.4, the Amdahl's law and actual speedup values evaluated in both the CNR-IREA cluster and AWS Cloud cases are shown.

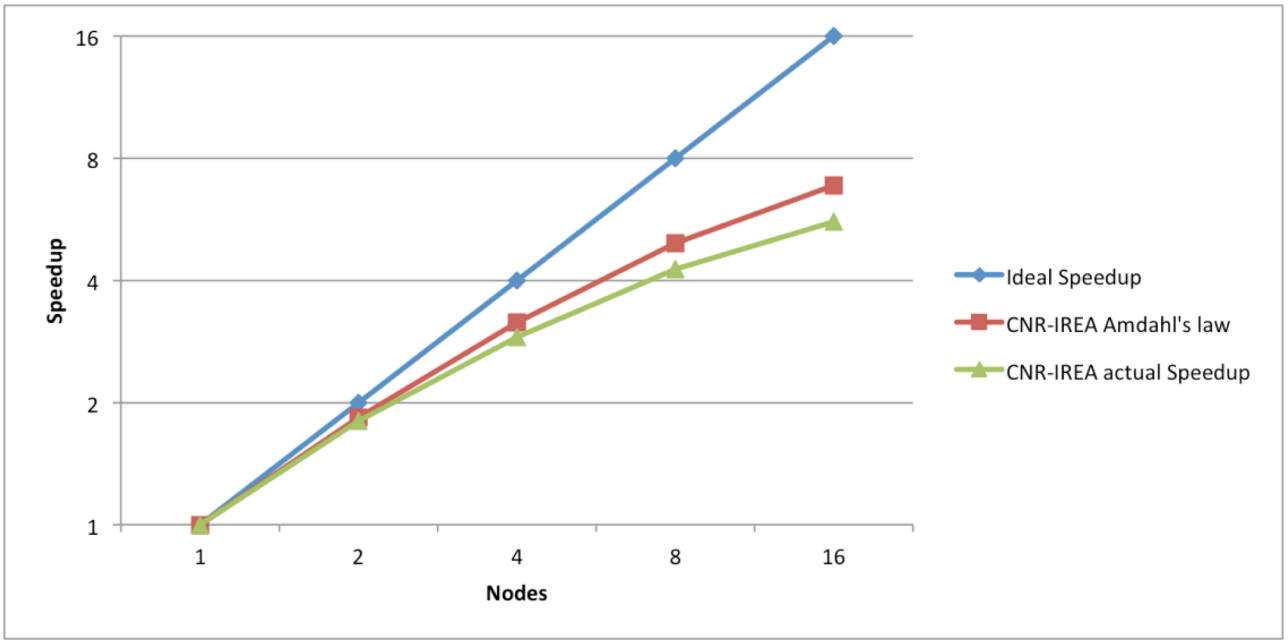


Figure 3.6: P-SBAS performances on CNR-IREA cluster: Speedup as a function of the number of processors N (green/triangles). The ideal achievable speedup (blue/diamonds) and the Amdahl's law behavior (red/squares) are also shown.

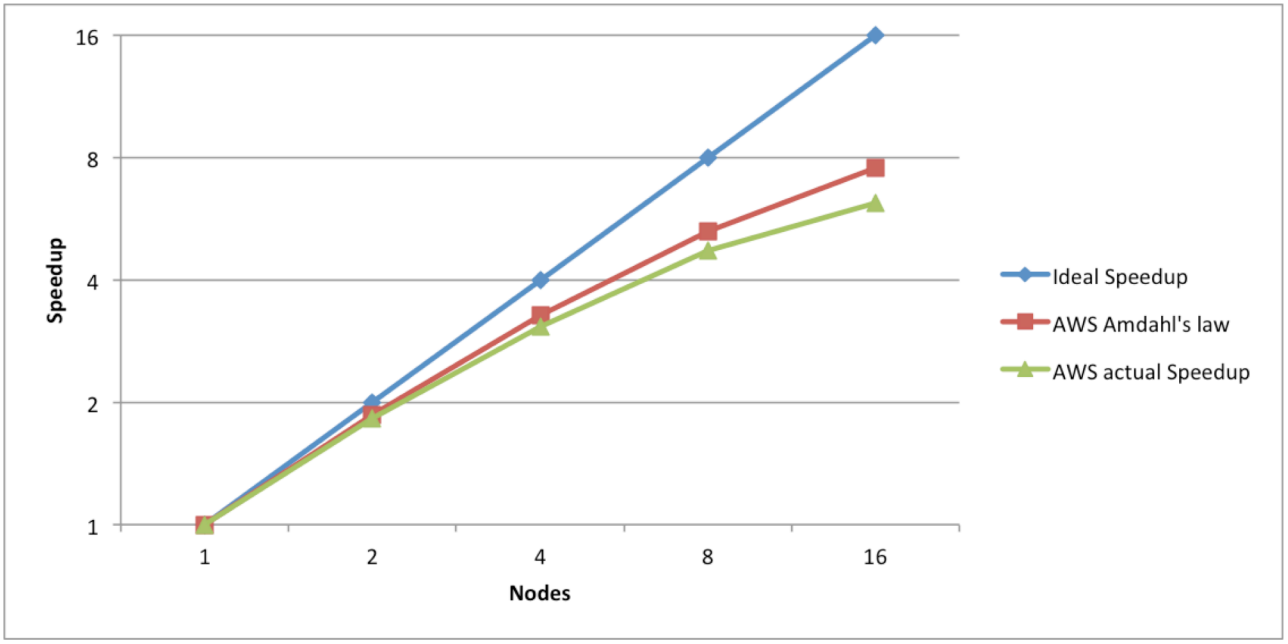


Figure 3.7: P-SBAS performances on AWS cloud: Speedup as a function of the number of processors N (green/triangles). The ideal achievable speedup (blue/diamonds) and the Amdahl's law behavior (red/squares) are also shown.

Furthermore, the percentage deviations between the theoretical (Amdahl's law) and experimental behavior are provided, quantitatively confirming the good match between the P-SBAS performance both on cluster and Cloud.

In order to identify which is the performance bottleneck when the number of parallel processes increases, some useful metrics, such as the read/write bandwidth and average queue length, provided by the AWS CloudWatch monitoring system [20] have been analyzed in detail.

It turned out that the loss of efficiency relevant to the 16-nodes processing is ascribable to the I/O workload as it understandably increases with the number of parallel processes, which concurrently read or write on the common storage volume. Hence, the factor that mainly lowers the P-SBAS scalable performance in our case is essentially the storage volume access bandwidth that is smaller than the network bandwidth (256 MB/s vs. 10 Gb/s).

In the following the performances of two steps of the P- SBAS algorithm, the image coregistration and the spatial phase unwrapping (blocks C and I of Figure 3.1), which are characterized by very different I/O workloads, are thoroughly analyzed. To this aim, we considered two metrics: the read/write bandwidth as well as the average length queue, both measured with respect to the EBS storage volume. The former metric is provided by AWS in KiB/s and has been converted in MB/s to be consistent with the known storage volume access bandwidth, while the latter metric represents the number of pending I/O requests and, therefore, it is a latency measure. For the sake of simplicity in Tables 3.6 and 3.7, only the metrics relevant to the processing carried out with 4, 8, and 16 nodes are reported, which are the most significant to understand how the I/O work-loads affects the scalable performances when the number of concurrent processes increases.

The image coregistration has been selected as it is one of the most demanding step of P-SBAS in terms of I/O workload; moreover, it shows the poorest scalable performances among all the P-SBAS algorithm steps. Indeed, accordingly to Table 3.6, the image coregistration presents an efficiency that

strongly degrades while the number of nodes increases. Even if this step could be liable to an improvement from a programming viewpoint in order to reduce its I/O workload, its behavior is helpful to correlate the inadequate scalable performances to the read/write bandwidth saturation. As a matter of fact, such a bandwidth is practically saturated for eight nodes as it already reaches the amount of 230 MB/s of data transfer; this value increases and becomes even 250 MB/s for 16 nodes.

The critical value of the average queue length for the employed storage configuration is around 40 counts, indeed this number depends on the I/O capacity of the selected EBS volume [20]. Once again eight nodes are already enough to reach the maximum allowed latency threshold and a greater number

Table 3.6

Image Coregistration Step: Efficiency and I/O metrics (retrieved from the CloudWatch monitoring system) as a function of the number of exploited nodes on AWS cloud

Image Coregistration	Elapsed Time [minutes]	Efficiency	Average Read/Write Bandwidth [MB/s]	Average Queue Length [count]
4 nodes	37	81%	125	20
8 nodes	23	65%	230	40
16 nodes	19	39%	250	50

Table 3.7

Phase Unwrapping Step: Efficiency and I/O metrics (retrieved from the CloudWatch monitoring system) as a function of the number of exploited nodes on AWS cloud

Phase Unwrapping	Elapsed Time [minutes]	Efficiency	Average Read/Write Bandwidth [MB/s]	Average Queue Length [count]
4 nodes	38	89%	40	7
8 nodes	21	80%	65	12
16 nodes	12	70%	75	20

of employed nodes would not bring a significant reduction in the elapsed time for this step.

On the contrary, the phase unwrapping step, even if very burdensome from a computational viewpoint, is less heavy for what concerns I/O operations. This step presents satisfactory scalable performances as shown in Table 3.7, with a 70% efficiency in correspondence with 16 nodes and it would therefore further scale if a greater number of nodes were used. Indeed, in this case, both the read/write bandwidth and the queue length values are evidently below the saturation threshold.

It is worth noting that the scalable performances of the P-SBAS processing chain in the presented Cloud configuration can be further improved by increasing the storage volume access bandwidth by configuring a RAID 0 striping with a greater number of "provisioned IOPS" volumes (up to 12). This would allow us to exploit a larger number of nodes without saturating the storage volume access bandwidth, as long as it is supported by an adequate network bandwidth, with the performance limit given by the Amdahl's law.

3.5.4 Cost analysis

As a counterpart to the time performances analysis, we hereafter present the study of the costs relevant to the P-SBAS runs which have been carried out on the AWS cloud by exploiting the two different resources configurations discussed beforehand. Such an analysis is very useful to evaluate the trade-off between times and costs of an interferometric SAR data processing depending on the exploited AWS computing resources and would allow a scientific user to select the best configuration according to its specific time/cost requirements. The overall cost of a P-SBAS run is given by the sum of the processing cost, i.e. the cost of the exploited EC2 instances, and the storage cost. The AWS on-demand EC2 instances are charged for hourly use, therefore the price depends on the hourly cost can change depending on the region from which instances are rented. In our case the presented costs are relevant

to the EU Ireland instances. The provisioned IOPS EBS volumes, instead, are charged by both the provisioned amount of GB per month, which is invariant for the same processing, as well as the provisioned IOPS multiplied by the percentage of days you provision for the month. Table 3.8 shows the costs relevant to the tests, which have been carried out over Napoli Bay.

In particular the m2.4xlarge hourly cost is 0.980 USD, while the c3.8xlarge cost is 1.680 USD. Moreover, the storage volume is charged by 0.138 USD per GB-month of provisioned storage and 0.072 USD per provisioned IOPS-month. The costs presented in Table 3.8 have been extracted from the monthly AWS invoice (V A T excluded). However, they can be easily estimated by exploiting the following formula:

$$C_{TOT} = C_{Instance} * \left[T_{TOT|hours} + (N-1) * T_{par|hours}\right] + C_{Storage}^{GB} + C_{Storage}^{IOPS} \quad (1)$$

where:

$$C_{Storage}^{GB} = C_{monthly\ GB} * \ GB_{provisioned} \quad (2)$$

$$C_{Storage}^{IOPS} = C_{monthly\ GB} * IOPS_{provisioned} * \frac{T_{TOT|days}}{30} \quad (3)$$

In the above formulas the symbol C indicates costs, T_{tot} and T_{par} and stand for the P-SBAS overall and parallel execution times, respectively, N represents the number of exploited nodes and the notations $(.)|hours$ or $(.)|days$ indicate times, which have to be rounded up to hours or days. In Figure 5.8 we show the graphs representing the times/costs trade-off of the experimental tests, which have been carried out on AWS cloud and presented in the previous paragraph. In particular, the P-SBAS

Table 3.8: AWS instances and storage costs relevant to the processing shown in Table 5.3

Nodes	m2.4xlarge costs (USD)	Single EBS provisioned IOPS costs (USD)	m2.4xlarge overall costs (USD)	c3.8xlarge costs (USD)	RAID configu-ration costs (USD)	c3.8xlarge overall costs (USD)
1	53	34,8	87,8	69	44,4	113,4
2	60	25,2	85,2	72,4	25,2	97,6
4	72	15,6	87,6	89	25,2	114,2
8	95	15,6	110,6	109	25,2	134,2
16	-	-	-	163	25,2	188,2

Performances, depending on the number of exploited EC2 instances, are represented in the time-cost plane for the m2.4xlarge configuration (red) and for the c3.8xlarge configuration (blue).

The graphs of Figure 3.8 allow an understanding of the best AWS resources configuration and the best number of nodes to exploit in order to process an ENVISAT standard dataset according to the specific time/cost requirements. In our case, since processing time reduction is a priority to provide results as soon as possible, the choice has been focused on the 8 nodes exploitation with either the m2.4xlarge or the c3.8xlarge configuration. The latter configuration presents a 21% costs increment and a 33% times decrease with respect to the m3.4xlarge one, therefore for the case study discussed in the following section we opted for the 8 nodes run with the c3.8xlarge instances and the RAID configured storage. It is worth noting that we do not include in this analysis the cost relevant to both the initial phase of cloud resources configuration and deployment, that is, essentially, the storage creation, and data upload as well as the final P-SBAS output download, since they are not really significant amounting to 5% of the overall cost.

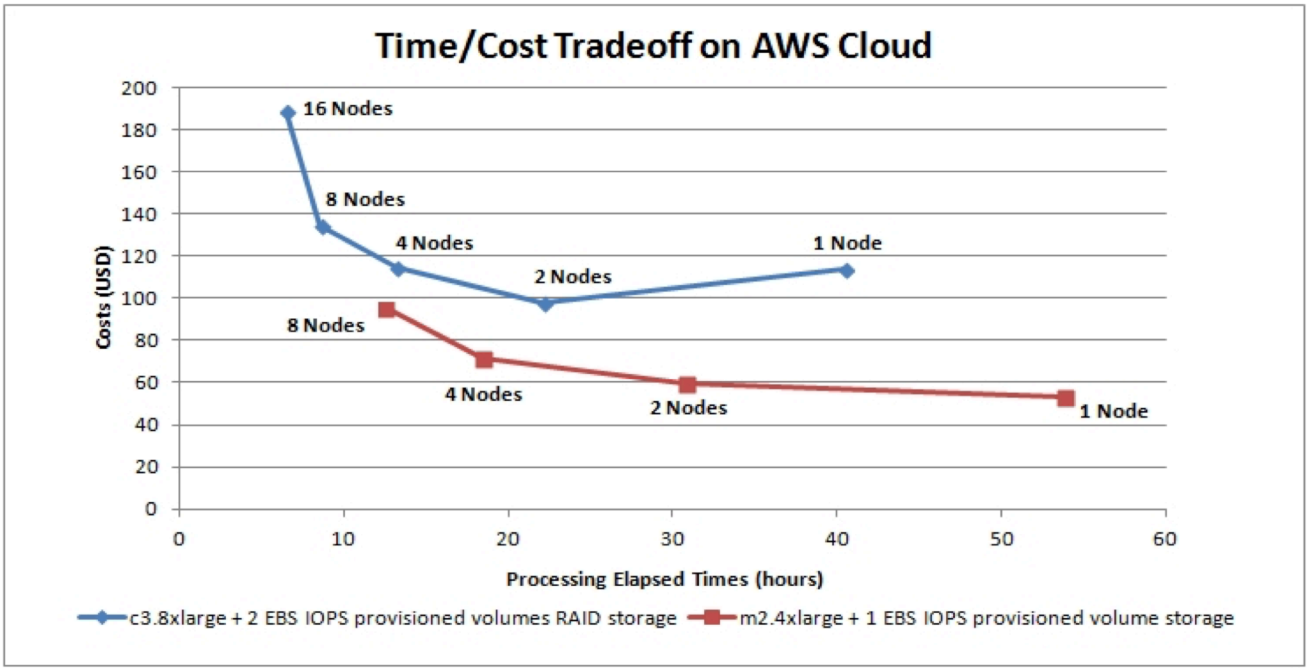


Figure 3.8. Trade-off between the elapsed times (horizontal axis) and the overall costs (vertical axis) relevant to the P-SBAS processing of the Napoli Bay area dataset, changing the number of exploited EC2 instances. The red line refers to the configuration with the m2.4xlarge instances and a single EBS volume storage; the blue line refers to the configuration with the c3.8xlarge instances and RAID configured storage with 2 EBS volumes.

Table 3.9
Additional costs relevant to the initial deployment phase and data transfer

Task	Time	Cost
Create the Amazon EBS volumes for your array	30 seconds	-
Attach the Amazon EBS volumes to the instance that you want to host the array	10 seconds	-
Pre-warm your volumes	2 hours	7.4 $
Initialize the logical RAID device from the newly attached Amazon EBS volumes	10 seconds	-
Create a file system on your RAID device	10 seconds	-
Create a mount point for your RAID array	5 seconds	-
Upload the data	.5 hours	1.85 $
Download final output	10 seconds	-

However, for the sake of completeness, in Table 3.9 the times and costs of all the steps required by the initial deployment phase and data transfer are listed. Moreover, we do not take into account the costs of IDL licenses that are provided by the CNR-IREA institution.

3.6 Large area analysis: the Southern California experiment

In this section we present the processing of a large interferometric data stack relevant to a portion of southern California that also includes the Los Angeles metropolitan area, in which a large variety of phenomena that induce ground displacements occur, such as active seismic faults, aquifer seasonal extraction and recharge, oil extraction, and agricultural fields. In particular, 172 ENVISAT SAR data acquired from ascending orbits, spanning the 2004–2010 time interval and covering an area of approximately 200x200 km^2 (corresponding to four slightly overlapped ENVISAT frames), have been considered. In this case the overall input data come to approximately 80 GBytes. The goal of such an experiment is to exploit the results that have been achieved in the previous section to process the aforementioned SAR dataset by properly adjusting cloud computing resources according to the P-SBAS requirements, thus proving that cloud environments allow dealing with large SAR data interferometric processing within short time scales and at limited costs. In particular, we simultaneously processed four ENVISAT frames through the deployment of four identical independent pools of resources interconnected by different subnets. The test bed encompasses the whole 32 instances, 8 for the processing of each ENVISAT frame, in the same configuration employed for the first set of experiments (one master node acting both as worker node and NFS server and seven worker nodes) and 4 storage volumes in a RAID 0 configuration, each one of 500 GBytes. In Table 3.10, times

Table 3.10
Execution Tests on Amazon cloud with c3.8xlarge instances (Southern California area)

Frame numbers	Data set (# images)	c3.8xlarge instances	Elapsed time (hours)	Cost* (USD)
1	47	8	16,7	242
2	44	8	15,5	227
3	43	8	11	192
4	38	8	10,5	192
Overall	**172**	**32**	**16,7**	**853**

*Note that the represented costs include both the instances and storage prices.

and costs related to each dataset processing are represented. Note that the elapsed times are different because they essentially depend on both the number of SAR dataset images and, above all, on the scene's electromagnetic characteristics that have a great impact on the number of the coherent pixels. In this case the overall data processing took less than 17 hours, with a total cost of $821 USD.

The costs that are shown in Table 3.10 include processing cost, storage cost, and additional deployment and data transfer cost. In Figure 3.9 the retrieved LOS mean deformation velocity map of the overall area covered by the ENVISAT frames is showed. This image has been realized by firstly geocoding the four deformation velocity maps relevant to each frame. They have been afterwards properly combined and superimposed on Google Earth. Such a simple post-processing phase has been carried out *a posteriori*. It is worth noting the various deformation patterns that are clearly shown in Figure 3.9. For some of these the temporal evolution of the measured displacements is represented through the time series plot included in Figure 5.9. In particular, Figure 5.9(b) is relevant to the maximum deformation zone of the Santa Ana basin while Figure 5.9(c) is relevant to the injection of fluids in the Downey area. Figure 5.9(d) is relevant to the deformation surrounding the Pomona fault while Figure 5.9(e)

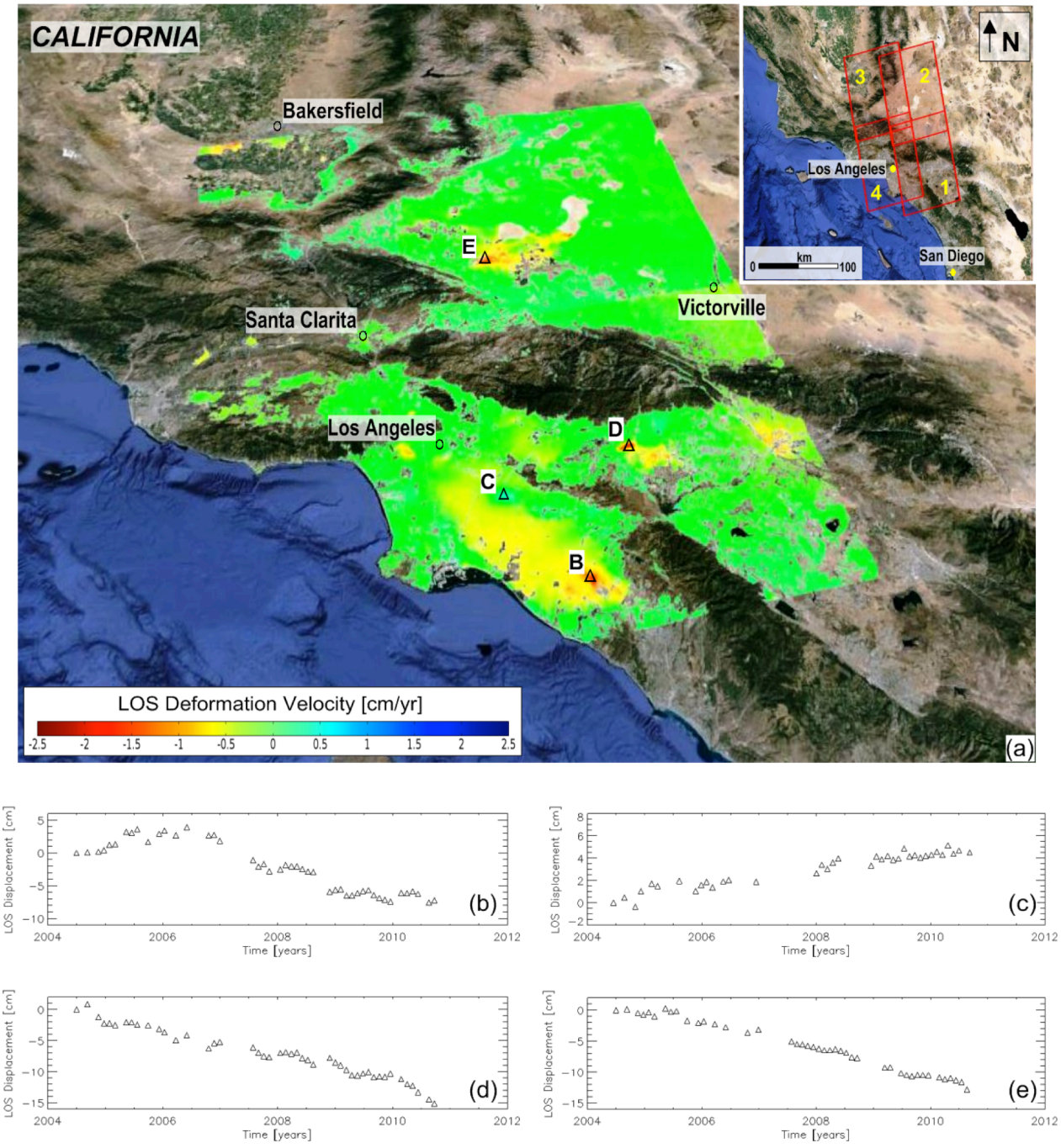


Figure 3.9 Mean deformation velocity map on the investigated region of Southern California. The plots of the retrieved deformation time series are reported in (b)-(c)-(d)-(e) for four selected zones identified in (a) as B-C-D-E, respectively. Note also that in the inset of (a) it is shown the location and the extension of the four ENVISAT frames exploited for this analysis (see Table 5.10).

represents the land subsidence movement in the Lancaster zone that is very likely caused by water withdrawal.

3.6 Summary

In this chapter, a case study concerning the migration to the AWS cloud computing environment of the advanced DInSAR algorithm referred to as P-SBAS, which generates interferometric products useful for the investigation of Earth surface deformation phenomena, was presented and discussed. The cloud deployment was carried out by encapsulating the overall P-SBAS application in virtual machines running on the Cloud; moreover, the cloud resources provisioning and configuration phases were implemented in an automatic way. Such an approach allowed us to preserve the intrinsic P-SBAS parallelization strategy and, therefore, to straightforwardly evaluate its performances within a cloud environment by comparing them with those achieved on an HPC in-house cluster. An experimental framework including two case studies was presented to envisage an effective roadmap for DInSAR processing of large data volumes in cloud computing environments. The first case study concerned the processing of the complete sequence of 64 SAR images of an ENVISAT frame (about 100x100 km2) acquired on the Naples Bay area (southern Italy), which has already been used as a benchmark for interferometric analysis. The experimental results showed that, within the AWS cloud, the P-SBAS processing presents very good performance in terms of processing elapsed times when the number of exploited nodes increases. Moreover, a thorough analysis of the P-SBAS processing costs when the type and the number of the exploited AWS cloud resources change was accomplished. Such a study allows evaluation of the more suitable configuration in accordance to the specific time/cost requirements, which for the considered experiment, proved to be the one set up by the c3.8xlarge instances and the storage in a RAID configuration. By exploiting the experimental results obtained in the first case study, we carried out the second experimental analysis of processing of a large interferometric dataset consisting of four ENVISAT frames, for a total of 172 images covering an area located in southern California of about 200x200 km^2, including the entire Los Angeles metropolitan

zone. More specifically, this analysis was performed by employing 32 EC2 c3.8xlarge instances; the overall P-SBAS processing took less than 17 hours with a cost of about USD 850.

References

[1] E. Sansosti, F. Casu, M. Manzo, and R. Lanari, "Space-borne radar inter- ferometry techniques for the generation of deformation time series: An advanced tool for Earth's surface displacement analysis," *Geophys. Res. Lett.*, vol. 37, 2010, doi: 10.1029/2010GL044379.

[2] P. Berardino, G. Fornaro, R. Lanari, and E. Sansosti, "A new algorithm for surface deformation monitoring based on small baseline differential SAR interferograms," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 11, pp. 2375–2383, Nov. 2002.

[3] F. Casu, M. Manzo, and R. Lanari, "A quantitative assessment of the SBAS algorithm performance for surface deformation retrieval from DInSAR data," *Remote Sens. Environ.*, vol. 102, pp. 195–210, 2006.

[4] A. Manconi *et al.*, "On the effects of 3-D mechanical heterogeneities at Campi Flegrei caldera, southern Italy," *J. Geophys. Res. Solid Earth*, vol. 115, 2010, doi: 10.1029/2009JB007099.

[5] R. Lanari, F. Casu, M. Manzo, and P. Lundgren, "Application of the SBAS-DInSAR technique to fault creep: A case study of the Hayward fault, California," *Remote Sens. Environ.*, vol. 109, pp. 20–28, 2007.

[6] F. Calò *et al.*, "Enhanced landslide investigations through advanced DInSAR techniques: The Ivancich case study, Assisi, Italy," *Remote Sens. Environ.*, vol. 142, pp. 69–82, 2014.

[7] R. Lanari, P. Lundgren, M. Manzo, and F. Casu, "Satellite radar interferometry time series analysis of surface deformation for Los Angeles, California," *Geophys. Res. Lett.*, vol. 31, 2004, doi: 10.1029/2004GL021294.

[8] M. Bonano, M. Manunta, A. Pepe, L. Paglia, and R. Lanari, "From previ- ous C-band to new X-band SAR systems: Assessment of the DInSAR mapping improvement for deformation time-series

retrieval in urban areas," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 1973–1984, Apr. 2013.

[9] A. Pepe, E. Sansosti, P. Berardino, and R. Lanari, "On the genera- tion of ERS/ENVISAT DInSAR time-series via the SBAS technique," *IEEE Geosci. Remote Sens. Lett.*, vol. 2, no. 3, pp. 265–269, Jul. 2005.

[10] M. Bonano, M. Manunta, M. Marsella, and R. Lanari, "Long-term ERS/ENVISAT deformation time-series generation at full spatial reso- lution via the extended SBAS technique," *Int. J. Remote Sens.*, vol. 33, pp. 4756–4783, 2012.

[11] S. Salvi *et al.*, "The Sentinel-1 mission for the improvement of the scien- tific understanding and the operational monitoring of the seismic cycle," *Remote Sens. Environ.*, vol. 120, pp. 164–174, May 2012.

[12] A. Rucci, A. Ferretti, A. M. Guarnieri, and F. Rocca, "Sentinel 1 SAR interferometry applications: The outlook for sub millimeter measure- ments," *Remote Sens. Environ.*, vol. 120, pp. 156–163, May 2012.

[13] F. De Zan and A. Monti Guarnieri, "TOPSAR: Terrain observation by progressive scans," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 9, pp. 2352–2360, Sep. 2006.

[14] F. Casu *et al.*, "SBAS-DInSAR parallel processing for deformation time- series computation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 8, pp. 3285–3296, Aug. 2014.

[15] P. Imperatore *et al.*, "Scalable performance analysis of the parallel SBAS- DInSAR algorithm," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS'14)*, Quebec City, Canada, pp. 350–353, 2014.

[16] S. Elefante *et al.*, "G-POD implementation of the P-SBAS DInSAR algo- rithm to process big volumes of SAR data," in *Proc. Conf. Big Data from Space (BiDS'14)*, 2014.

[17] P. Rosen, K. Shams, E. Gurrola, B. George, and D. Knight, “InSAR scientific computing environment on the cloud,” in *Proc. AGU Conf.*, San Francisco, CA, USA, 2012 [Online]. Available: http://fallmeeting.agu.org/2012/eposters/eposter/in31c-1508/

[18] S. Elefante *et al.*, “SBAS-DInSAR time series generation on cloud computing platforms,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS’13)*, Melbourne, Australia, pp. 274–277, 2013.

[19] M. Hansen *et al.*, “High-resolution global maps of 21st-century forest cover change,” *Science*, vol. 342, pp. 850–853, 2013.

[20] Amazon. (2014). *Amazon Web Services* [Online]. Available: http://aws. amazon.com/

[21] Google. (2014). *Google Earth Engine* [Online]. Available: http://www. google.it/intl/it/earth/outreach/tools/earthengine.html

[22] HNX. (2015). *Helix Nebula Marketplace* [Online]. Available: http://hnx. helix-nebula.eu/

[23] K. Yelick, S. Coghlan, B. Draney, and R. S. Canon, “The Magellan report on cloud computing for science,” U.S. Dept. Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR), Washington, DC, USA, Dec. 2011.

[24] O. Terzo, L. Mossucca, A. Acquaviva, F. Abate, and R. Provenzano, “A cloud infrastructure for optimization of a massive parallel sequencing workflow,” presented at the 12th IEEE/ACM Int. Symp. Cluster Cloud Grid Comput. (CCGrid’12), Ottawa, Canada, 2012, pp. 678–679.

[25] G. Franceschetti and R. Lanari, *Synthetic Aperture Radar Processing*, Boca Raton, FL, CRC,Mar. 1999.

[26] E.Sansosti, P. Berardino, M. Manunta, F. Serafino and G. Fornaro, "Geometrical SAR image registration," *IEEE Trans. Geosci. Remote Sens.,* vol. 44, no. 10, pp. 2861–2870, Oct. 2006.

[27] A. Pepe and R. Lanari, "On the extension of the minimum cost flow algorithm for phase unwrapping of multitemporal differential SAR interferograms," *IEEE Trans. Geosci. Remote Sens*., vol. 44, no. 9, pp. 2374–2383, Sep. 2006.

[28] V. Andrikopoulos, T. Binz, F. Leymann and S. Strauch, ″How to Adapt Applications for the Cloud Environment,″ *Computing,*Springer, vol. 95, Issue 6, pp. 493-535,2013.

[29] Exelis V isual Information Solution. A vailable: http://www.exelisvis.com

[30] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh and B. Lyon, "Design and Implementation of the SUN Network Filesystem," *USENIX Conference Proceedings*, 1985

[31] Esteban Molina-Estolano, Maya Gokhale, Carlos Maltzahn, John May, John Bent, and Scott Brandt. 2009, "Mixing Hadoop and HPC workloads on parallel filesystems," *Proceedings of the 4th Annual Workshop on Petascale Data Storage* (PDSW '09).

[32] John J. Rehr , Fernando D. Vila , Jeffrey P. Gardner, Lucas Svec , Micah Prange "Scientific Computing in the Cloud," *IEEE Computing Now*, Vol. 12, No. 3, pp. 34-43, 2010

[33] P. Barham, B. Dragovic, K. Fraser, S. Hand, Steven, T. Harris, H. Alex, R.Neugebauer, I. Pratt, Ian and A. Warfield, "Xen and the Art of Virtualization,"*SIGOPSOper. Syst. Rev.*, pp. 164-177, December 2003.

[34] G. Hager and G. Wellein. *Introduction to High Performance Computing for Scientists and Engineers*. CRC Press, 2010.

[35] H. El-Rewini and M. Abd-El-Barr, *Advanced Computer Architecture and Parallel Processing*, John Wiley & Sons, Inc, 2005.

[36] I. Foster, *Designing and Building Parallel Programs*. USA: Addison Wesley, 1995.

[37] HELIX NEBULA - The Science Cloud. Available: http://www.helix-nebula.eu.

Chapter 4

A New Cloud Computing Solution for an Efficient Implementation of the P-SBAS Approach

We present in this chapter an efficient cloud computing (CC) implementation of the Parallel Small BAseline Subset (P-SBAS) algorithm, which is an advanced Differential Interferometric Synthetic Aperture Radar (DInSAR) technique for the generation of Earth surface displacement time series through distributed computing infrastructures (both multi-node and multi-core). In particular, the proposed approach consists in distributing data and processing among the different computing nodes, so that each one can concurrently work on data that are physically stored on the local volume. Such a rationale allows minimizing the data transfer and the network load to improve the P-SBAS efficiency and scalability within CC environments.

The proposed P-SBAS implementation has been extensively validated on two real case scenarios by exploiting the Amazon Web Services (AWS) Elastic Cloud Compute (EC2) resources. The former experimental analysis involves the processing of a large (128 SAR images) COSMO-SkyMed dataset and is aimed at demonstrating the P-SBAS scalable performances. The latter allows us to show the P-SBAS capability to generate DInSAR results at a regional scale (150,000 km^2 in Southern California) in very short time frames (about 9 hours), by simultaneously processing 18 ENVISAT frames that correspond to a total of 741 SAR images exploiting in parallel 144 AWS computing nodes.

The presented results confirm the effectiveness of the proposed P-SBAS CC solution in processing large amount of data, thus contributing to extend the frontiers of DInSAR investigation at a very large scale.

4.1 SAR Big Data Scenario

Considering this fast evolution that is characterizing the DInSAR technology during the last 20 years, rapidly pushing towards a Big Data scenario, the development of effective solutions able to properly deal with the storage, the accessibility, the transfer and, above all, the processing of such a huge SAR data flow is strongly needed.

Within the framework of the advanced DInSAR processing technicques, a parallel algorithmic solution for the SBAS approach, referred to as P-SBAS, which implements the complete advanced DInSAR processing chain (from raw data focusing up to the displacement time-series and deformation mean velocity maps generation) and is able to exploit distributed computing architectures, has been recently developed [27]. This solution has also been implemented within the European Space Agency (ESA) Grid Processing on Demand (G-POD) environment to make available the SBAS technique for on-demand processing [28].

However, in house or grid resources can be a bottleneck due to their limitedness. This is particularly the case if the provisioning of multi-user services, allowing worldwide scientists to process SAR data through advanced DInSAR techniques and to access to the achieved results, is envisaged. In this scenario, a key role can be played by the Cloud Computing (CC) technologies that are well established within the Information and Communication Technologies (ICT) field, with important spin-offs also for the scientific application context [29]-[31]. Note that the exploitation of customized computing infrastructures built up within CC environments can be crucial in many respects; first, because of the practically unlimited computing facilities they make available; second, for the flexibility they provide, which allows extensive resources optimization. In such a context, some studies on the P-SBAS algorithm migration to the public CC environment Amazon Web Services (AWS), have already been carried out [33], [34]. These works concerned the evaluation of the P-SBAS parallel performances

achievable on a cloud platform and allowed highlighting the main issues affecting the P-SBAS scalability, which are related to the bottlenecks that occur when huge data volumes, which are processed and generated by the algorithm, need to be transferred and shared among the different computing nodes. In particular, very high network and Input/Output (I/O) capabilities are required to sustain good scalable performances, but, when the number of computing nodes exploited for the parallel computation increases, these resources inevitably tend to saturate [33], [34].

In this work we present an efficient CC implementation of the P-SBAS algorithm that has been designed to overcome the above mentioned scalability limitations encountered within the previous implementation [33], [34] already described in the chapter 3, permitting to achieve good scalable performances without requiring very high performing resources in terms of I/O throughput. The rationale of this novel implementation lies in distributing the data storage involved in the P-SBAS processing among different nodes connected through the NFS protocol [35], instead of centralizing it, thus parallel handling the overall I/O workload. Such a strategy, however, because of the complexity of the P-SBAS processing chain, required the design and the development of an ad-hoc management of the data-flow dependencies, as well as a proper job scheduling to handle the entire P-SBAS workflow.

In order to investigate the scalable capability of the proposed solution, we carried out a thorough experimental study by exploiting the AWS Elastic Cloud Compute (EC2) public cloud platform [36]. In particular, we performed two kind of analysis; the former, aimed at evaluating the scalable performances achieved with the P-SBAS NFS distributed storage implementation, has been developed in continuation of the outcomes presented in [33] and [34]. In this case the obtained results show how the new implementation leads to performance improvements even without exploiting machines with very high performing features and capabilities, thus also permitting a strong reduction of the costs relevant to the P-SBAS processing (cheaper resources can be employed). The latter analysis, besides, has been targeted to show the results of a large scale DInSAR processing experiment involving 741

ENVISAT acquisitions that has been performed through the presented P-SBAS NFS distributed storage implementation by exploiting a big number of computing nodes (144 instances) provided by AWS environment. The presented results confirm the effectiveness of the proposed CC solution of the P-SBAS processing chain.

4.2 P-SBAS NFS-based Distributed Storage Implementation

4.2.1 Rationale

In this section we discuss a CC implementation for the P-SBAS algorithm allowing the advanced DInSAR processing of very large SAR datasets, such as those provided by the COSMO-SkyMed sensors and, in the near future, by the Sentinel-1 satellites. We remark that, although CC environments offer the possibility of using a practically infinite number of computing elements, a working-well and cost-effective solution can be achieved only if the resources utilization is maximized. Consequently, an efficient CC implementation should exhibit a good scalability in the achieved computing performances, when the number of employed computing nodes increases. This is not a trivial issue because, as also demonstrated by the available literature on this subject [29], [43]-[49] the achievement of high scalable performances on CC platforms is a demanding task, especially for scientific applications involving huge data volumes.

The P-SBAS algorithm, deals with an incessant processing of large data time series that are shared among different computing nodes. A P-SBAS scalable performance analysis, performed exploiting the AWS Elastic Compute Cloud (EC2) and concerning a 64 ENVISAT images dataset, has been thoroughly discussed in [33], [34] and in chapter 3. This analysis allowed us also to highlight the main issues affecting the P-SBAS scalability, which are related to the exploitation of a centralized common

storage disk, shared via a Network File System (NFS) among the different computing nodes. Indeed, the concurrent access to such a disk from the parallel jobs of the P-SBAS algorithm, running on different nodes, makes both the data transfer and the I/O workload to increase proportionally with the number of exploited nodes. This means that, though the network and the disk access bandwidth can be high performing, there will always be a number of parallel jobs leading to saturation. Therefore, the centralized NFS storage solution, although easily solving the P-SBAS data dependency issue, represents a bottleneck. Indeed, due to technological limitations, it sets a restriction to the maximum number of nodes that can be exploited and, therefore, to the overall algorithm scalability.

A potential solution to overcome this issue could be represented by employing a parallel file system, as this technology is ad-hoc designed to handle a high number of concurrent jobs that simultaneously deal with large files. However, in a CC environment, the use of a parallel file system will inevitably burdensome the network load. Consequently, the requirement of a high network bandwidth will be necessarily needed, thus providing once again a bottleneck.

Accordingly, we propose a P-SBAS CC solution based on distributing the I/O workload among the different computing nodes, instead of centralizing it on a unique storage. Such a new solution has been designed to minimize the I/O and data transfer among different nodes in order to achieve good scalable performances without necessarily requiring extremely powerful resources. Moreover, it also allows us to limit the costs relevant to the resources utilization within CC environments because it permits to choose cheap machines.

To realize the above-mentioned solution we implemented an architecture in which each node is directly attached to a local disk, on which it performs its part of I/O operations; this permits to comply the requirement of providing high data throughput fully benefiting from the I/O performances of each disk. Moreover, to ensure data transfer minimization, a proper job scheduling has been designed in order to guarantee that the parallel processes of the P-SBAS algorithm will mostly deal with files that are

physically stored on their local disks. This strategy will ensure that the network workload is strongly reduced. Obviously, although significantly cut down, a data transfer among different nodes is still required in some steps of the P-SBAS algorithm and, therefore, all the computing nodes of the implemented architecture need to be mutually connected via the NFS. Such a kind of data distribution, therefore, relies on saving the data as whole files, but spreading them among the storage volumes of different computing nodes, which are attached through the NFS protocol. The implemented strategy allows simultaneously speeding up the I/O operations and reducing the network usage, eliminating both the disk access and the network bottlenecks.

A detailed description of both the implemented computing architecture and the adopted scheduling strategy of the proposed P-SBAS CC solution is provided in the following paragraphs.

4.2.2 Computing Architecture

This paragraph is aimed at describing the computing architecture that we have designed to implement the previously mentioned P-SBAS NFS-based distributed storage CC solution.

The overall architecture has been realized by exploiting the AWS CC services, which provide on-demand delivery of IT resources with a pricing policy based on “pay what you use” [36].

We first implemented an Amazon Virtual Private Cloud (VPC) [44] that is a logically isolated section of the AWS in which resources can be launched in a completely defined virtual private network (VPN). Second, we configured a customized Amazon Machine Image (AMI) containing the operating system (we chose the Fedora distribution but different selection can be carried out), the AWS Command Line Interface (CLI) tool [46], which allows controlling multiple AWS services from the command line and automate them through scripts, and the software needed for the P-SBAS processing, such as IDL,

Fortran and C. Moreover, since the IDL software needs a license file to run, it has been packed directly within such an AMI, which is mounted on all the computing nodes of the implemented platform.

Figure 4.1 depicts the computing architecture that we built up, which consists of one master node and N worker nodes. The master node runs the entire P-SBAS workflow by managing the scheduling of the parallel jobs among the worker nodes. Moreover, the master node is configured as a network gateway to interface the implemented virtual private network outward, and it also performs computation.

All the nodes (both master and workers) are connected each other via NFS, therefore each node is at

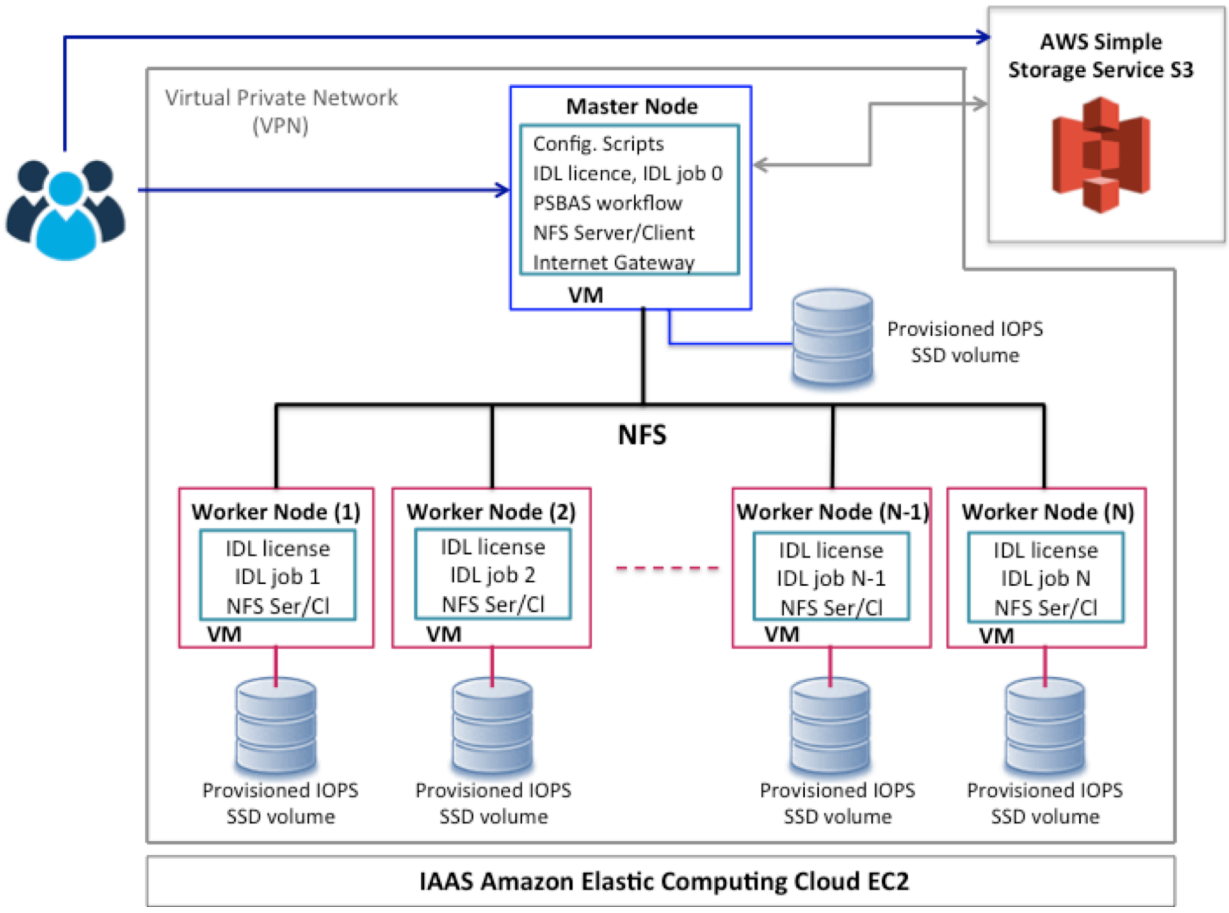


Figure 6.1: Computing architecture implemented within the AWS public cloud for the NFS-based distributed storage implementation of the P-SBAS approach. Note that the architecture consists of one master node and several worker nodes (each of these with attached its own local disk), which are all connected via the NFS protocol.

the same time NFS server and client.

Finally, each computing node has got a local disk (or, more generally, a dedicated storage volume) that is directly attached to it, thus ensuring high I/O performances. Such local disks are maintained active only during the computation lifetime, indeed, at the end of the P-SBAS processing, the final results are transferred to a permanent storage (i.e., AWS Simple Storage Service S3 [47]) and the local disks are deleted when the computing nodes are turned off.

It is worth noting that the creation of the overall computing architecture is automatically carried out through some scripts that we developed by exploiting both Bash programming language and the AWS CLI tool. The whole computing architecture creation phase lasts few minutes independently of the number of machines that are launched, because the resources are allocated in parallel.

The final results transfer time is of the order of minutes if only time series and mean deformation velocity maps are considered; on the contrary, if also intermediate products such as interferogram stacks are required, it could last up to one hour depending on the amount of data that is concerned.

4.2.3 Scheduling strategy

The design of the scheduling strategy of the P-SBAS NFS-based distributed-storage CC solution is not a trivial task. Indeed, as described in Chapter 1, the P-SBAS algorithm is characterized by the succession of parallel and intrinsic sequential steps that, in some points of the processing chain, need to use all the output data resulting from the previous operations. Different strategies concerning the jobs scheduling and the storage policies have been, therefore, implemented to successfully port the P-SBAS application in a CC environment using the novel approach of partitioning data among different computing nodes.

Let us now consider the storage policy that is organized by following three different modalities:

a) The data are saved in a unique volume, which is the master's local disk. This solution is implemented when small files are generated and this condition usually occurs during the sequential steps, as files containing data parameters and information useful for the whole processing are produced.

b) The data are stored in specific storage volumes mutually connected via NFS, which have been pre-determined by the job scheduler. This modality is employed when the data needs to be processed by a single node and therefore the data will be stored in its corresponding hard-disk.

c) The data are stored evenly and randomly across the nodes' volumes. This solution is implemented when the specific data need to be initially distributed among the computing nodes. Therefore it is only carried out at the beginning of the processing.

Following the above discussion on the implemented scheduling and storage solutions, let us now focus on the full description of the overall processing chain.

First of all, the input raw data are stored in the EC2 S3 persistent storage. At this stage, the job scheduler will randomly but evenly copy such data to the computing nodes storage volumes. The raw data ingestion is realized in a parallel mode and is fully locally performed in the sense that each computing node will extract only the data that have been physically copied on its hard-disk.

The focusing step (block A of Figure 2.1), which generates the Single Look Complex (SLC) SAR images, is executed in a parallel mode. As previously explained, the raw data have already been extracted and stored in the different storage volumes. Hence, each computing node will implement the focusing operation on the data that are physically present on its volume as, in this way, a minimal data transfer through the network is guaranteed. The common parameters that are needed to perform this step through will be retrieved by the different computing nodes accessing the master's disk via NFS.

The DEM conversion in SAR coordinates step (block B of Figure 2.1) occurs in parallel mode and the overall data are currently stored in a single volume.

The co-registration step (block C of Figure 2.1) can be executed only after that the SAR image identified as reference (Master acquisition) has been chosen. The NFS mutual sharing ensures that every single computing node can access the disk containing all the information relevant to the master image to be able to execute the co-registration.

The image registration refining and the interferogram generation steps (blocks E and G of Figure 2.1) involve the processing of an interferometric images pair. In this case the job scheduling has been designed to ensure that a minimal data transfer would occur. To achieve this task, if the two images of an InSAR pair are both stored in the same disk, the node of this volume will process the images. Conversely, in the case in which the images are saved in different volumes, the job scheduling policy will assign the processing pair to one of the node in which one image is stored. This job scheduling occurs in a dynamical manner in the sense that during the processing, as soon as a node becomes free, the image pair is assigned.

The step that follows is the unwrapping one that is very demanding in terms of CPU usage and I/O operations. The sequential step of the temporal phase unwrapping (block H of Figure 2.1) is designed to write the output data (which are large 3D matrices containing phase information) directly on the disks on which the spatial phase unwrapping (block I of Figure 2.1) will be, afterwards, executed. This solution will ensure that the spatial phase unwrapping will make use of a minimal network bandwidth. The reduce step is performed in a sequential manner and its output data are saved in the master node storage.

The final step concerning the displacement time series generation (block L of Figure 2.1) is executed in parallel but it stores the final matrices containing the displacement time series and the mean deformation velocity information on the master node storage volume.

4.3 Experimental results

In this section we evaluate the scalable performances of the P-SBAS distributed storage implementation showing the improvement that is achieved with respect to the centralized storage solution [34], particularly when the number of nodes that are exploited for the computation increases. Moreover, the following analysis is aimed at demonstrating that the presented P-SBAS implementation shows good parallel performances without needing to exploit very high network and disk access bandwidth performances, thus allowing reducing both the processing times and costs.

4.3.1 Computational platforms

The computing architectures that we exploited for the comparison between the P-SBAS CC centralized [33], [34] and the presented distributed storage implementations have been built up exploiting the AWS EC2 resources located in the Ireland region. Among the available EC2 instances, we selected those allowing us to demonstrate how the new implementation overcomes the network bottleneck problem, which characterized the centralized storage implementation discussed in [33] and [34], leading also to a significant cost reduction. In particular, instead of exploiting instances with the maximum network bandwidth available within EC2 (i.e, 10 Gb/s), which were needed in [33] and [34] to reach a satisfactory scalability, we chose machines with a network performance referred to as “high”, which is around 1 Gb/s. This reduction for the requested network bandwidth permitted us to save the 42% of the costs related to the instance payment.

More precisely, for our experimental analysis, both as master and worker nodes, we selected the m4.4xlarge machines because they have got characteristics satisfying the P-SBAS requirements in terms of RAM capacity and processor (number of cores), equivalently to [33] and [34], but with the

above-mentioned 1 Gb/s network capability and with a configuration optimized for the Amazon storage volumes (Elastic Block Store, EBS). This means that these instances have dedicated connections to the storage disks to which are attached. In Table 4.1 the characteristics of the m4.4xlarge instances are shown in details.

Note that, throughout our analysis, we refer to an instance as to a single computing node.

Moreover, for the centralized storage implementation, as a shared common storage volume we selected an IOPS (Input Output Operation per Second) provisioned SSD storage volume, ensuring very high I/O performances. In particular, we configured the volume with 16000 IOPS, which correspond to 250 MB/s disk access bandwidth, which is the maximum throughput achievable by the m4.4xlarge instances.

Table 4.1
AWS Instance Type Configuration

	m4.4.xlarge
Architecture	64 bit
Processor	Intel Xeon E5-2676 v3
vCPU	16
RAM	64 GB
Network	High (around 1Gb/s)

The volume size depends on the overall amount of data produced by the P-SBAS processing and this storage volume has been attached to the master node. Clearly, the storage configuration cannot disregard the network capability because, during the parallel steps of the P-SBAS processing, the data to be read and written on the common storage are transferred via network; consequently, the actual I/O bandwidth is given by the minimum between the disk access and the network bandwidth. In our

experiments we intentionally set the disk I/O capacity to its maximum in order to highlight that the network becomes the bottleneck.

Concerning the distributed storage implementation, instead, each computing node of the implemented platform had attached its own storage volume. In this case we exploited an IOPS provisioned SSD storage volume with 1000 IOPS for each worker node, and 4000 IOPS for the master node, as this latter one, in several steps of the P-SBAS processing, performs more I/O operations than the other nodes. In these experiments we carefully set the I/O performances strictly necessary to sustain the I/O workload of each computing node. Moreover, the disk sizes have been selected so that the total amount of data exploited and used by the P-SBAS algorithm could be uniformly distributed among them.

4.3.2 Parallel Computing Metrics and Exploited SAR Dataset

In order to quantitatively evaluate the scalable performances of the implemented P-SBAS processing chain, appropriate metrics have been adopted. Let *N* be the number of computing nodes used to solve a problem and T_1 the execution time of the sequential implementation to solve the same problem, the speedup S_N of a parallel program with parallel execution time T_N is defined as follows [50], [51]:

$$S_N = \frac{T_1}{T_N} \tag{1}$$

Accordingly, the speedup is a metric that compares the parallel and sequential execution times. Moreover, to quantitatively assess the effect of the serial parts of the algorithm on the attainable speedup, the well-known Amdahl's law is hereafter considered [51]

$$S_N = \frac{1}{f_S + \frac{1 - f_S}{N}} \quad (2)$$

where $f_S \in [0,1]$ is the parallel program fraction that has been sequentially executed (sequential fraction) [51]. It is also worth mentioning that the formulation (6.2) of Amdahl's law does not take into account either the load unbalancing or the data transfer overhead.

For our experimental analysis we exploited a COSMO-SkyMed interferometric dataset acquired over the Napoli bay, a volcanic and densely urbanized area in Southern Italy that includes the active caldera of Campi Flegrei, the Vesuvius volcano, and the city of Napoli. The choice of using this COSMO-SkyMed dataset is due to the fact that, currently, both in terms of image size and temporal sampling of the sensor acquisition, it is a good example of large InSAR datasets within the remote sensing scenario. In particular, the overall dataset is composed of 128 acquisitions (for a total amount of about 130 GBytes, concerning the raw data) relevant to ascending orbits, covering an area of about 60x60 km2 and spanning the 2012-2014 time interval.

The selected dataset has been processed by using the implemented P-SBAS algorithm in order to generate the DInSAR products. The amount of intermediate and final results produced during our processing is of about 4.5 TBytes.

4.3.3 Scalable performance analysis

This section is aimed at evaluating and comparing the scalable performances of the developed P-SBAS NFS-based distributed storage (DS) implementation with those based on the centralized storage (CS) implementation [33], [34]. In particular the goal is to show how the distributed storage solution presented in this work preserves the scalability when the number of computing nodes increases,

differently to the centralized storage one, thus ensuring that the P-SBAS processing times that are achieved within the considered CC environment, are significantly improved.

We carried out the scalability analysis by running the P-SBAS algorithm within the AWS CC environment with 8, 16, and 32 m4.4xlarge instances (as said, one instance corresponds to one computing node) both with the centralized and the distributed storage implementations. Moreover, we performed the sequential run (with only one computing node), which essentially coincides for the two implementations, in order to evaluate the Amdahl's law (see (4.2)). The total processing time of the P-SBAS sequential execution is 478,8 hours and, starting from this time, we computed the sequential fraction of the processing, which represents the 3,7%. We did not execute the 2 and 4 nodes runs because they would have not been significant for the objective of our analysis, which is the investigation of the P-SBAS performances when the number of computing nodes increases, i.e. when maintaining the scalability becomes critical [33], [34].

In Table 4.2 the P-SBAS processing times relevant to the above-described experiments are summarized.

Table 4.2
P-SBAS Processing Times

Parallel fraction of P-SBAS processing				**Overall P-SBAS processing**		
N nodes	**CS. parallel fraction processing times [h]**	**DS. parallel fraction processing times [h]**	**Processing times reduction**	**CS. total processing time [h]**	**DS. total processing time [h]**	**Processing times reduction**
8	60,8	57,6	**5,2 %**	79	75	**5 %**
16	37,2	33,1	**14,5 %**	55	51	**7,2 %**
32	27,5	17,6	**38,5 %**	45	35,3	**21,5 %**

In particular, we reported both the elapsed times relevant to the parallel fraction of the P-SBAS processing, which actually contain the information about the achieved scalability, and the times of the

overall P-SBAS processing, which include also those relevant to the sequential part of the algorithm. It is worth noting that the fifth and the sixth columns are the total elapsed times computed from the data extraction until the deformation map generation, including the time to transfer the obtained deformation map to the EC2 S3 persistent storage. Note also that the time to transfer the initial raw data to the master node is not included in the table; it depends, indeed, on the external network that is used and in our case it was of about half an hour. When only the parallel part is considered, Table 4.2 shows that the new P-SBAS implementation leads to a noticeable reduction of the elapsed time when it is compared to the use of the centralized solution. A reduction of 5% is achieved when 8 nodes are considered but this amount significantly grows reaching a value of 38% when 32 nodes are taken into account. Such a performance improvement increases when the number of nodes working in parallel increases, because of the progressive saturation of the overall system in terms of bandwidth and disk access relevant to the centralized storage implementation, which does not occur in the distributed storage case. It is important to remark that an elapsed time reduction can be observed only during the parallel steps execution, as the new implementation has no effect on the sequential steps.

Moreover, Figure 4.1 shows the Amdahl's law behavior (black line), which is the reference of the maximum achievable theoretical performance, and the actual speedups that have been experimentally measured, for the P-SBAS centralized and distributed storage implementations, shown in green and red columns respectively.

In Table 4.3 the values of the represented speedups are also given. Comparing the two graphs and the relevant values in Table 4.3, it is evident that, in the case of the CS implementation (green line), the actual speedup begins to slightly diverge from the Amdahl's law already in correspondence with the 8 nodes run and it progressively gets worse moving to 32 nodes, as the network bandwidth saturation grows when the number of exploited computing nodes increases. On the contrary, in the case of the DS implementation (red line), the actual speedup follows quite well the Amdahl's law behavior, thus

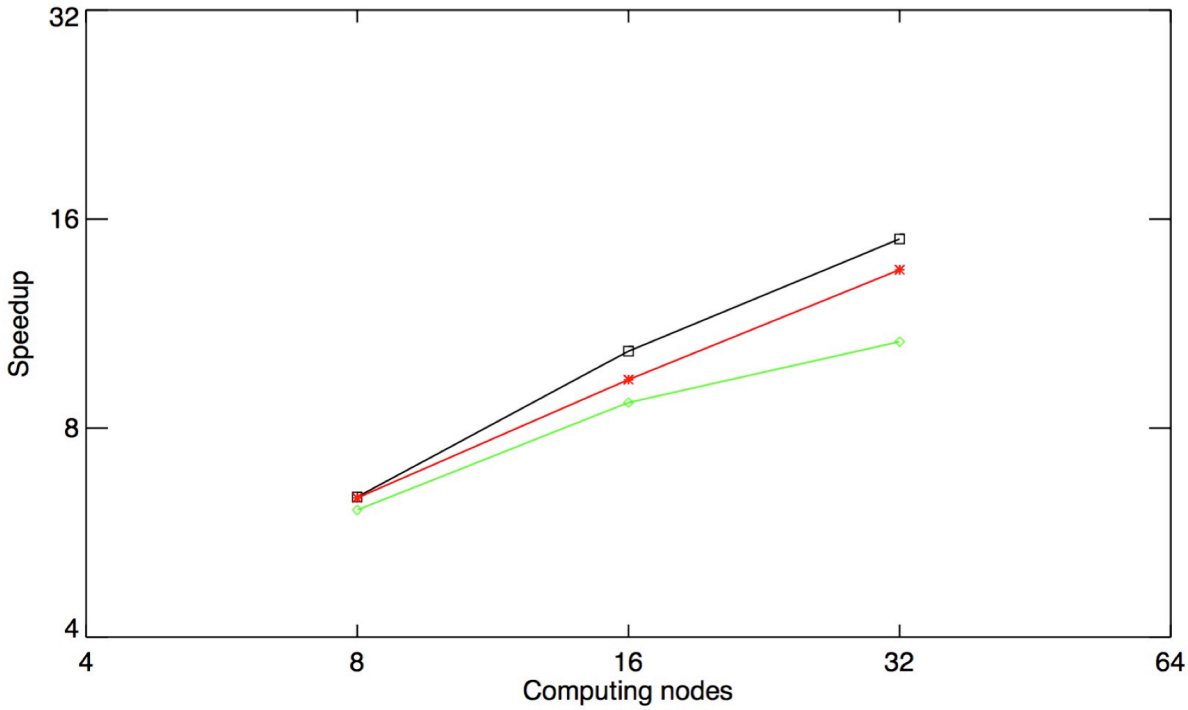


Figure 4.1: Speedup analysis with 8,16 and 32 computing nodes. The Amdhal's law behavior (black line) and the speedups relevant to the NFS P-SBAS centralized storage (green line) and distributed storage (red line) implementations are depicted

exhibiting the good scalability that is fulfilled. It is worth emphasizing that, since this new implementation is specifically designed to minimize the P-SBAS algorithm bottleneck, such a good scalability is maintainable when the number of exploited computing nodes further increases.

Finally, in Table 4.4, we present the costs of the P-SBAS processing for the described experiments. It is evident that, in all the considered cases, the P-SBAS distributed storage implementation exhibits lower costs with respect to the centralized storage one, thus confirming that the novel solution allows not only preserving the scalability and cutting down the processing times, but also saving money.

In Figure 4.2, the generated mean deformation velocity map is shown; note that it has been geocoded and afterward superimposed on a multilook SAR image of the investigated area. We remember that the estimated mean deformation velocity has been only computed in coherent areas; accordingly, areas in which the measurement accuracy is affected by decorrelation noise have been excluded from the map.

Table 4.3
P-SBAS Speedup analysis with 8,16 and 32 computing nodes

N nodes	Amdahl's law behavior	NFS P-SBAS distributed storage Speedup	NFS P-SBAS centralized storage Speedup
8	6.37	6.35	6.1
16	10.32	9.39	8.71
32	14.98	13.37	10.65

Table 4.4

P-SBAS Processing cost with 8,16 and 32 computing nodes

N nodes	CS Implementation overall processing costs [$]	DS Implementation overall processing costs [$]	Percentage Cost Reduction between CS and DS
8	930	839	9,8 %
16	1197	1052	12,1 %
32	1869	1548	17,2 %

It is worth noting in Figure 4.2 a significant deformation pattern corresponding to the area of the Campi Flegrei caldera. Moreover, the computation of the temporal evolution of the detected deformation has also been carried out for each coherent point of the scene. We represent in the inset of Figure 4.2 the chronological sequences of the computed displacement of a point located in the maximum deformation area of the Campi Flegrei caldera [52], [53].

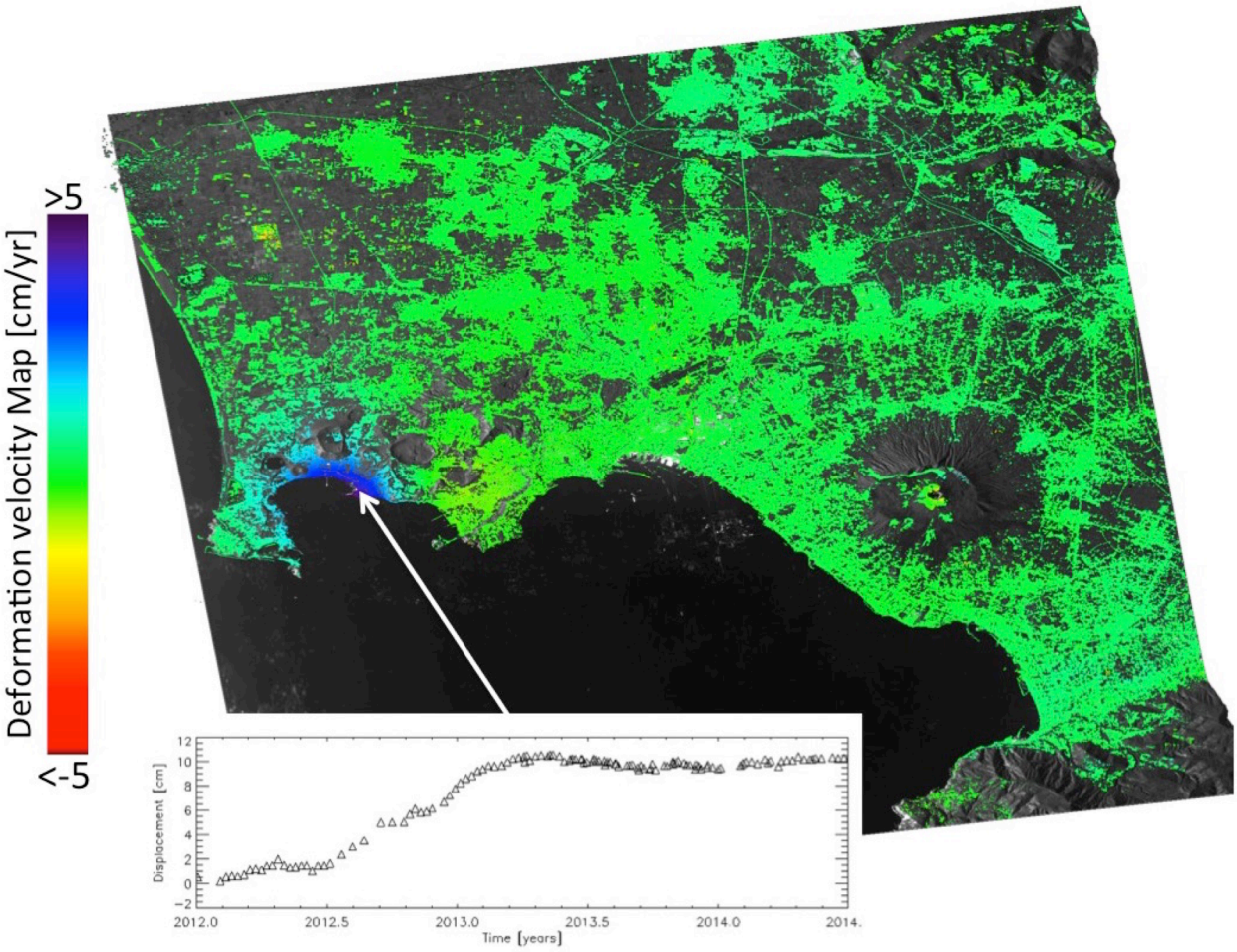


Figure 4.2: Mean deformation velocity map of the Napoli Bay area, generated via the P-SBAS algorithm on the AWS cloud. The inset shows the temporal analysis of a single pixel in a zone affected by maximum deformation pattern

4.4 A region-scale P-SBAS processing case study: the ENVISAT analysis over California

We show in this section a region scale DInSAR analysis performed through the presented P-SBAS NFS-based DS implementation. Such an analysis has been carried out by processing in parallel 18 ENVISAT frames acquired over an area of the California region including both the San Diego and Los Angeles metropolitan areas, as well as the San Joaquin Valley region, as shown in Figure 6.3. This zone is characterized by a large variety of phenomena inducing ground displacements, such as active

seismic faults, aquifer seasonal extraction and recharge, oil extraction, and agricultural fields. In particular, the exploited dataset is composed of 741 ENVISAT SAR scenes acquired from ascending orbits, spanning the 2003-2010 time interval and covering an area of approximately 150,000 km^2. The overall input datasets (raw data) size is of about 560 GBytes.

To perform such a big processing we used 144 AWS instances running concurrently. In particular we exploited 8 instances for each ENVISAT frame processing, i.e., we carried out in parallel the processing of each frame on a subset of 8 computing nodes. During the raw data ingestion step of P-SBAS, some operations of selection and merging of the input raw data are performed in order to extract the correct initial datasets for the following of the processing.

In Figure 4.3 we depicted the spatial coverage of the 18 exploited ENVISAT frames; for each one of them we also indicated the number of images that have been effectively used for the processing after the raw ingestion step. Note that each of the considered ENVISAT frames is composed on average of about 40 images, the smaller one comprising 22 images, the larger one 52.

We run all the 144 instances as a part of a unique VPN, connecting them through the NFS protocol. One of the instances was configured as a global master node and managed both the scheduling and the supervision of the 18 concurrent ENVISAT frames processing among all the other instances. Moreover, each one of the subsets of 8 computing nodes that processed a single frame had its own local master to manage the parallel jobs of P-SBAS among its 8 instances. The global master node periodically checked the state of completion of all the frames processing and, as soon as one of them finished, moved the relevant results to the AWS S3 storage and turned off the corresponding instances.

Since an ENVISAT image is (about four times, in terms of pixels) smaller than a COSMO-SkyMed one, we could exploit in this case instances less performing with respect to those used for the experiment performed in the previous section, both in terms of RAM capacity and of CPUs' number.

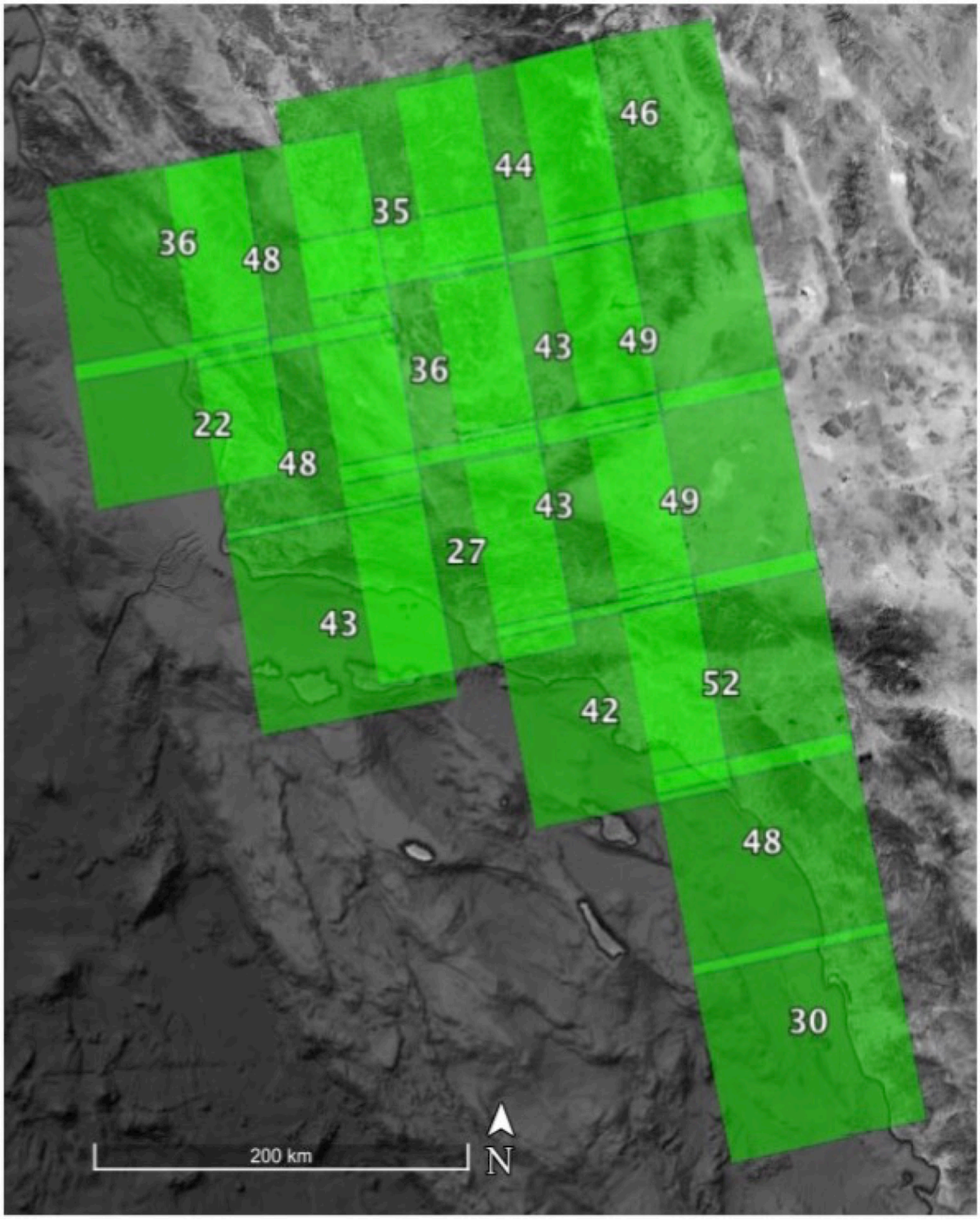


Figure 4.3: Image of the Footprints of the 18 ENVISAT frame over California area

More specifically, we selected the EC2 m4.2xlarge [36] instance for all the computing nodes except the global master node, for which a more powerful instance has been chosen (i.e., the c4.8xlarge).

Indeed, the global master not only had to periodically access all the other computing nodes to schedule the overall processing jobs as well as monitor its progress status, thus needing a greater network bandwidth, but it also performed computation as a local master of its subset of computing nodes. Note that the characteristics of the m4.2xlarge and c4.8xlarge instances are compared in Table 4.5.

Table 4.5
AWS Instance Type Configuration

	m4.2xlarge	**c4.8.xlarge**
Processor	Intel Xeon E5-2676 v3	Intel Xeon E5-2666 v3
vCPU	8	36
RAM	32 GB	60 GB
Network	High (around 1Gb/s)	10 Gigabit

Table 4.6
741 ENVISAT Images Processing Times and Costs

	Time
Maximum P-SBAS processing time between 18 ENVISAT frames	≈ 8 hours
Average P-SBAS processing time between 18 ENVISAT frames	≈ 6 hours
Data transfer from S3 storage to the EC2 instances	≈ 40 minutes
Overall architecture creation and configuration	10 minutes
Time to transfer main results to the S3 storage	5 minutes
Overall Experiment Cost	**< 1000 USD**
Average cost per ENVISAT frame	**≈ 55 USD**

Finally, for the local storage disks attached to the exploited instances we used 144 Provisioned IOPS SSD volumes each one equipped with 1000 IOPS, for a total of 144000 Provisioned IOPS.

In Table 4.6 we summarize the computing times of the described experiment: note that the overall experiment required less than 9 hours. The creation, i.e. the parallel launch of the 144 EC2 instances, and the whole architecture configuration phases took about 10 minutes to complete.

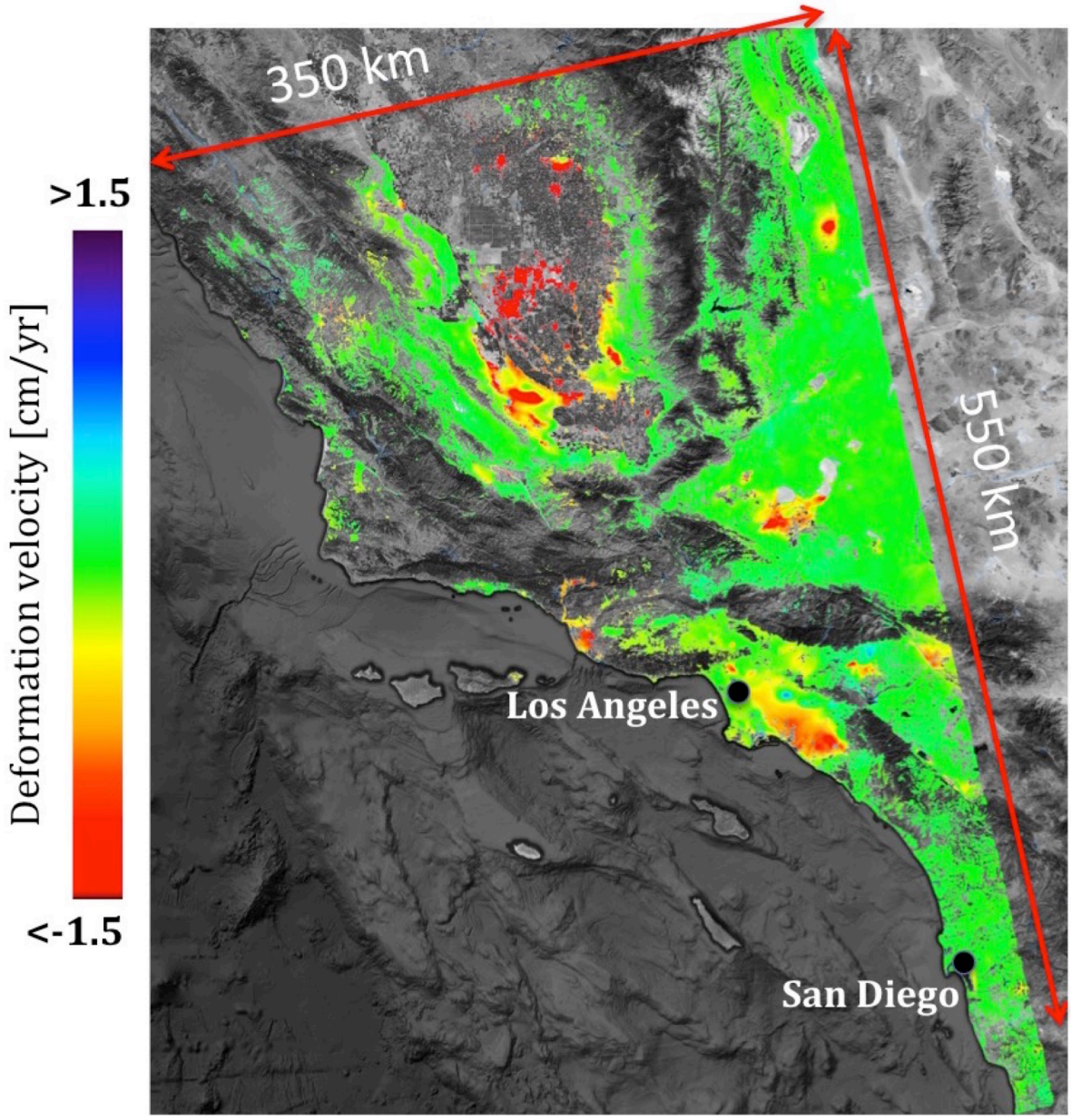


Figure 4.5: Mean deformation velocity map relevant to the area of California that has been analyzed

Then, the input data were moved from the AWS S3 archive to the EC2 instances. In particular, the 18 ENVISAT frames datasets were transferred in parallel to the computing nodes designated as local master for the corresponding processing. Such a transfer lasted about 40 minutes.

These actions were automatically carried out through the mentioned Linux Bash scripts that exploit the AWS CLI commands. The P-SBAS processing of all the 18 frames, i.e. for the 741 SAR scenes, lasted 8 hours. The average processing time per frame was about 6 hours; moreover, the shortest runtime was little more than 3 hours, the longest one 8 hours. The overall cost of the experiment was less than 1000 USD, with an average cost per frame of approximately 55 USD.

Clearly, the different P-SBAS runtimes depend on the size of the considered frame, in terms of number of acquired images, as well as on the electromagnetic characteristics of the ground scene illuminated by the sensor that determine the number of coherent pixels to be processed in several steps of the P-SBAS processing chain.

In Figure 4.3 the obtained final mean deformation velocity map is shown. This map has been achieved by joining the 18 mean deformation velocity maps obtained through the P-SBAS processing of the corresponding ENVISAT frames with a very simple post-processing phase that has been accomplished off-line.

4.5 Summary

We presented an efficient implementation of the P-SBAS DInSAR algorithm within a CC environment, which allows the generation of Earth's surface deformation time series and mean velocity maps starting from SAR data sequences. Such a novel implementation has been conceived to ensure good scalable performances within a CC environment, where a very large number of computing elements can be employed for the parallel SBAS processing. This is, indeed, a time-sensitive issue within the current

remote sensing scenario characterized by the availability of very huge SAR data archives whose full exploitation through advanced DInSAR methods requires appropriate solutions in terms of data storage, transfer and, above all, processing.

In particular, the proposed P-SBAS CC implementation has been designed to overcome the scalability bottlenecks, related to the original centralized NFS storage solution [27], [34], which are due to the intensive I/O workload of the P-SBAS algorithm when very large input datasets are concerned. The adopted rationale consists in distributing the data generated and exploited throughout the P-SBAS processing among the different storage volumes attached to the computing nodes working concurrently, which are connected via the NFS protocol. Moreover, the main goal has been to minimize the data transfer between different nodes, so that the network does not saturate. Such a strategy, however, because of the complexity of the P-SBAS processing chain, required the design of an *ad-hoc* management of the data-flow dependencies, as well as a proper job scheduling to handle the entire P-SBAS workflow.

A thorough experimental survey has been carried out to show the effectiveness of the novel P-SBAS CC implementation, by exploiting the Amazon Web Services EC2 public cloud platform. In particular, we carried out two kinds of analysis. The former has been aimed at evaluating the scalable performances achieved with the novel P-SBAS NFS distributed storage implementation and has been conducted by exploiting a large COSMO-SkyMed interferometric dataset composed of 128 SAR images. The obtained results show that the presented P-SBAS CC solution exhibits a very good scalabilty by exploiting up to 32 AWS instances (for a total of 512 virtual CPUs) with a noticeable improvement with respect to the previous solution based on a NFS centralized storage. Moreover, the proposed implementation allows us to exploit computing resources with moderate network and I/O capabilities in order to fulfill such a good scalability, thus strongly reducing the required costs.

In the latter analysis we showed the results of the DInSAR processing relevant to an extended area performed through the presented P-SBAS CC solution. In this case we processed in parallel 18 ENVISAT frames, corresponding to 741 SAR raw data, acquired over an area of the California region of 150,000 km^2, including both the Los Angeles and San Diego metropolitan areas, as well as the San Joaquin Valley region, which are interested by several phenomena inducing ground deformation. For this processing we exploited in parallel 144 AWS EC2 instances, each one with its own storage volume attached, for a total of 1152 CPUs and approximately 11 TBytes of storage. The overall experiment lasted approximately 9 hours with a cost of less than 1000 USD.

References

[1] R. Burgmann, P. A. Rosen, and E. J. Fielding, "Synthetic aperture radar interferometry to measure Earth's surface topography and its deformation, " *Annu. Rev. Earth Planet. Sci.*, vol. 28, pp. 169–209, May 2000.

[2] D. Massonnet and K. L. Feigl, "Radar Interferometry and its application to changes in the Earth's surface," *Rev. of Geophys.*, vol. 36, pp. 441–500, 1998.

[3] D. Massonnet, M. Rossi, C. Carmona, F. Adragna, G. Peltzer, K. Feigl, and T. Rabaute, "The displacement field of the Landers earthquake mapped by radar interferometry," *Nature*, vol. 364, no. 6433, pp. 138– 142, Jul. 1993.

[4] K. Mogi, "Relations between the eruptions of various volcanoes and the deformations of the ground surfaces around them," *Bulletin of the earthquake research institute,* vol. 36, pp. 99-134, 1958.

[5] M.A.Chinnery, "The deformation of the ground around surface faults," *Bulletin of the Seismological Society of America*, vol. 51, pp. 355-372, 1961.

[6] W.E. Farrell, "Deformation of the Earth by surface loads," *Reviews of Geophysics*, Vol. 10, Issue 3, pp. 761–797, August 1972.

[7] Y. Okada, "Surface deformation due to shear and tensile faults in a half-space," *Bulletin of the seismological society of America*, Vol.75, No. 4, pp. 1135-1154, August 1985.

[8] P.A. Hsieh,"Deformation-induced changes in hydraulic head during ground-water withdrawal," *Groundwater*, Vol. 34, pp. 1082-1089, 1996.

[9] R.F. Yerkes and R.O.Castle, "Surface deformation associated with oil and gas field operations in the United States," *Land subsidence*, 1969, pubs.rsc.org.

[10] F.M. Orr,Jr., "CO 2 capture and storage: are we ready?," *Energy& Environmental Science*, Vol. 2, pp. 449-458, March 2009.

[11] S.Rackley, *Carbon capture and storage*, 2009, books.google.com.

[12] C.Q.C.S.M.Shixiang and H. Dade, "Numerical Simulation Study of Ground Deformation in Underground Mining," *Metal Mine*, 2004.

[13] M.P. O' Reilly and B.M. New, "New Settlements above tunnels in the United Kingdom-their magnitude and prediction," *Publication of: Institution of Mining and Metallurgy*, pp. 173-181, 1982, trid.trb.org.

[14] R.W. Clough and R.J., "Woodward Analysis of embankment stresses and deformations, "*Journal of Soil Mechanics & Foundations Div*, 1967

[15] G. Franceschetti and R. Lanari, *Synthetic Aperture Radar Processing*, Boca Raton, FL, CRC,Mar. 1999.

[16] H.A. Zebker and J. Villasenor, "Decorrelation in interferometric radar echoes," *IEEE Trans. Geosci. Remote Sens.*, vol. 30, pp. 950–959, Sept. 1992.

[17] D. Massonnet, P. Briole, and A. Arnaud, "Deflation of Mount Etna monitored by spaceborne radar interferometry," *Nature*, vol. 375, no. 6532, pp. 567–570, Jun. 1995

[18] G. Peltzer, and P.A. Rosen, "Surface displacement of the 17 May 1993 Eureka Valley earthquake observed by SAR interferometry," *Science*, vol. 268, pp. 1333-1336, June 1995.

[19] Y. Fialko, M. Simons, and D. Agnew, "The complete (3-D) surface displacement field in the epicentral area of the 1999 Mw 7.1 Hector Mine earthquake, California, from space geodetic observations," *Geophys. Res. Lett.*, vol. 28, pp. 3063-3066, Aug. 2001.

[20] Sansosti, E., F. Casu, M. Manzo, and R. Lanari (2010), Space-borne radar interferometry techniques for the generation of deformation time series: An advanced tool for Earth's surface displacement analysis, Geophys. Res. Lett., 37, L20305, doi:10.1029/2010GL044379.

[21] P. Berardino, G. Fornaro, R. Lanari, and E. Sansosti, "A new algorithm for surface deformation monitoring based on small baseline differential SAR interferograms," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 11, pp. 2375–2383, Nov. 2002.

[22] Lanari R, Mora O, Manunta M, Mallorqui JJ, Berardino P, Sansosti E (2004). A small-baseline approach for investigating deformations on full-resolution differential SAR interferograms. IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, vol. 42, p. 1377-1386, ISSN: 0196-2892, doi: 10.1109/TGRS.2004.828196.

[23] M. Bonano, M. Manunta, M. Marsella, and R. Lanari, "Long-term ERS/ENVISAT deformation time-series generation at full spatial resolution via the extended SBAS technique," *Int. J. Remote Sens.*, vol. Vol. 33, no. February 2012, pp. 4756–4783, 2012.

[24] M. Bonano, M. Manunta, A. Pepe, L. Paglia and R. Lanari, "From Previous C-Band to New X-Band SAR Systems: Assessment of the DInSAR Mapping Improvement for Deformation Time-Series Retrieval in Urban Areas," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 1973–1984, Apr. 2013.

[25] S.Salvi, S. Stramondo, G.J. Funning, A. Ferretti, F. Sarti and A. Mouratidis, "The Sentinel-1 mission for the improvement of the scientific understanding and the operational monitoring of the seismic cycle,"*Remote Sens. Environ.*, vol. 120, pp. 164–174, May 2012.

[26] Copernicus - The European Earth Observation Programme. Available : http://www.copernicus.eu.

[27] Casu, F.; Elefante, S.; Imperatore, P.; Zinno, I.; Manunta, M.; De Luca, C.; Lanari, R., "SBAS-DInSAR Parallel Processing for Deformation Time-Series Computation," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of* , vol.7, no.8, pp.3285,3296, Aug. 2014.

[28] ESA Grid Processing on Demand. Available at: https://gpod.eo.esa.int.

[29] Gupta, A.; Faraboschi, P.; Gioachin, F.; Kale, L.V.; Kaufmann, R.; Lee, B.-S.; March, V.; Milojicic, D.; Suen, C.H., "Evaluating and Improving the Performance and Scheduling of HPC Applications in Cloud," *Cloud Computing, IEEE Transactions on* , doi: 10.1109/TCC.2014.2339858.

[30] John J. Rehr , Fernando D. Vila , Jeffrey P. Gardner, Lucas Svec , Micah Prange "Scientific Computing in the Cloud," *IEEE Computing Now*, Vol. 12, No. 3, pp. 34-43, 2010

[31] S. Hardman,A. Riofrio, K. Shams, D. Freeborn, P. Springer and B.Chafin, "Enabling earth science through cloud computing," *Aerospace Conference, 2012 IEEE*, vol., no., pp. 1-8, 3-10 March 2012
doi: 10.1109/AERO.2012.6187356.

[32] P. Rosen, K. Shams, E. Gurrola, B. George and D. Knight, "InSAR Scientific Computing Environment on the Cloud,"*AGU Conference*, San Francisco, 3-7 December 2012.

[33] Zinno, I., Mossucca, L., Elefante, S., De Luca, C., Casola, V., Terzo, O., Casu, F., Lanari, R., "Cloud Computing for Earth Surface Deformation Analysis via Spaceborne Radar Imaging: a Case Study," *IEEE Trans. Cloud Computing*, 2015, in press

[34] Zinno, I., Elefante, S., Mossucca, L., De Luca, C., Manunta, M., Terzo, O., Lanari, R., Casu, F., "A first Assessment of the P-SBAS DInSAR Algorithm Performances within a Cloud Computing Environment," *IEEE JSTARS*, 2015, 10.1109/JSTARS.2015.2426054.

[35] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh and B. Lyon, "Design and Implementation of the SUN Network Filesystem," *USENIX Conference Proceedings*, 1985.

[36] Amazon EC2. Available: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html.

[37] ESA - Sentinel Online. Available: https://sentinel.esa.int/web/sentinel/home.

[38] De Zan, F.; Monti Guarnieri, A.M.; TOPSAR: Terrain Observation by Progressive Scans Geoscience and Remote Sensing, IEEE Trans. on Volume 44, Issue 9, Sept. 2006 pp. 2352 – 2360.

[39] Sansosti, E.; Berardino, P.; Manunta, M.; Serafino, F.; Fornaro, G., "Geometrical SAR image registration," in Geoscience and Remote Sensing, IEEE Transactions on, vol.44, no.10, pp.2861-2870, Oct. 2006.

[40] A. Pepe and R. Lanari, "On the extension of the minimum cost flow algorithm for phase unwrapping of multitemporal differential SAR interferograms," *IEEE Trans. Geosci. Remote Sens*., vol. 44, no. 9, pp. 2374–2383, Sep. 2006.

[41] Costantini, M., "A novel phase unwrapping method based on network programming," in Geoscience and Remote Sensing, IEEE Transactions on , vol.36, no.3, pp.813-821, May 1998.

[42] Imperatore, P.; Pepe, A.; Lanari, R., "Multichannel Phase Unwrapping: Problem Topology and Dual-Level Parallel Computational Model," in Geoscience and Remote Sensing, IEEE Transactions on , vol.53, no.10, pp.5774-5793, Oct. 2015.

[43] Gupta, Abhishek, and DejanMilojicic. "Evaluation of HPC applications on cloud." *Open Cirrus Summit (OCS), 2011 Sixth.* IEEE, 2011.

[44] Amazon Virtual Private Cloud VPC. Available: https://aws.amazon.com/it/documentation/vpc/.

[45] Fedora. Available: https://getfedora.org.

[46] AWS Command Line Interface CLI. Available: http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html.

[47] AWS Simple Storage Service S3. Available: https://aws.amazon.com/s3/details/.

[48] Sadooghi, I.; Hernandez Martin, J.; Li, T.; Brandstatter, K.; Zhao, Y.; Maheshwari, K.; Pais Pitta de LacerdaRuivo, T.; Timm, S.; Garzoglio, G.; Raicu, I., "Understanding the Performance and

Potential of Cloud Computing for Scientific Applications," *Cloud Computing, IEEE Transactions on*, doi: 10.1109/TCC.2015.2404821

[49] Roloff, E.; Diener, M.; Carissimi, A.; Navaux, P.O.A., "High Performance Computing in the cloud: Deployment, performance and cost efficiency," *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, vol., no., pp.371,378, 3-6 Dec. 2012.

[50] H. El-Rewini and M. Abd-El-Barr, Advanced Computer Architecture and Parallel Processing, 2005.

[51] G. Hager and G. Wellein, Introduction to High Performance Computing for Scientists and Engineers, 2010.

[52] A. Pepe, E. Sansosti, P. Berardino and R. Lanari, "On the Generation of ERS/ENVISAT DInSAR Time-Series Via the SBAS Technique," *IEEE Geosci. Remote Sens. Lett.,* Vol. 2, No 3, pp. 265–269, 2005.

[53] M. Bonano, M. Manunta, M. Marsella, and R. Lanari, "Long-term ERS/ENVISAT deformation time-series generation at full spatial resolution via the extended SBAS technique," *Int. J. Remote Sens.*, vol. Vol. 33, no. February 2012, pp. 4756–4783, 2012.

Chapter 5

Development of a Web Service for the Unsupervised Retrieval of Earth's Surface Deformation

The knowledge of Earth's surface deformation provides key insights into phenomena of great interest not only for the scientific community but also for regional and national authorities. Terrain displacement measurements are, indeed, very important to effectively investigate natural events such as earthquakes, volcano unrests and landslides, but also to detect, monitor and possibly mitigate the impact of anthropogenic activities such as ground water exploitation, oil and gas extraction, mining, etc. Nowadays, Differential SAR Interferometry (DInSAR) is one of the most used remote sensing techniques for the investigation of Earth's surface deformation phenomena [1,2]. It permits the retrieval of surface deformation maps with centimetre to millimetre accuracy, starting from the phase difference (interferogram) of SAR image pairs relevant to the same area of interest but acquired at different epochs and with a significantly small orbital spatial separation (baseline) [3,4]. Originally, DInSAR was developed to analyze single deformation episodes, such as earthquakes [5], but it has subsequently evolved towards the study of temporal behavior of the detected displacements. This evolution has been possible thanks to (a) the availability of temporally extended SAR data archives and (b) the implementation of the so-called "multi-temporal" (also referred to as "advanced") DInSAR algorithms [6–14] that permit the generation of surface deformation time series (and the corresponding mean velocity maps) of an observed area through the exploitation of a sequence of interferograms. Among several, a largely used multi-temporal DInSAR technique is the one referred to as the Small BAseline Subset (SBAS) [10]. The SBAS approach relies only on small spatial and temporal baseline

interferograms, which define a system of equations that can be inverted to retrieve the deformation time series, through a method based on Singular Value Decomposition (SVD) [15]. Note that, thanks to the above mentioned small baseline constraint, it is possible to drastically mitigate the noise phenomena affecting the DInSAR interferograms, referred to as decorrelation effects [16], thus maximizing the number of reliably analyzed SAR pixels. Recently, a parallel implementation of the SBAS algorithm [17], referred to as P-SBAS, which allows the generation of deformation time series in a very short time by taking advantage of distributed computing systems, has been presented.

The DInSAR results can be achieved at a relative low cost by benefiting from the free availability of the SAR archives acquired by the European satellites (ERS, ENVISAT, Sentinel-1) during the last 25 years, and from the large disposal of open access software for DInSAR data processing [18–20]. However, to obtain high quality multi-temporal DInSAR results, an extensive and non-trivial processing has to be carried out on the SAR data. This often limits the access to such a technology by non-expert users, who do not necessarily want to become skilled on DInSAR data processing, while they are typically more attracted by the analysis and interpretation of the retrieved displacement results.

In this chapter we present a tool, which is available via the web, aimed at generating in an unsupervised way surface deformation mean velocity maps and time series through the P-SBAS algorithm. Such a tool is implemented within the ESA's Grid Processing on Demand (G-POD) environment [21], which allows for operational web processing services based on federated computing facilities. The direct access of the G-POD environment to the ESA SAR data Virtual Archive 4 [22], together with a user-friendly interface, allows us to set up an efficient and on-demand P-SBAS processing web tool addressed to scientists that are less expert on interferometric SAR data processing.

5.1 G-POD Environment

G-POD is part of the Geohazards Thematic Exploitation Platform (GEP) activity originated by ESA [24]. The GEP, formerly named the Supersites Exploitation Platform (SSEP), was originally specifically designed for Earth Observation (EO) data exploitation in the context of the Geohazard Supersites and Natural Laboratories [25]; however, it has been subsequently expanded in order to address a broader field of objectives concerning the CEOS Pilots on Seismic Hazards and Volcanoes [26]. In particular, G-POD is an environment that was designed by ESA to both orchestrate and manage data processing workflows, in which applications aimed at exploiting EO data can be easily and directly plugged in.

More specifically, a generic application can be encapsulated within such a virtual environment and can exploit both distributed high-performance processing resources and large volumes of archived data, in order to provide the scientific community with new EO services [27]. G-POD benefits from the access to the ESA computing facilities as well as to their EO data archives, and provides a friendly web user interface that permits the processing of jobs on a distributed computing system, rather than on PCs or in-house user's workstations as for traditional applications. This also implies for the final user:

- no installation and update efforts, service access from everywhere (through a web browser) and, at any time;
- no need to download the input data;
- no need to procure and maintain expensive processing hardware.

The G-POD environment is, indeed, a complex distributed architecture that is constituted of different logical subsystems, such as computing facility, web portal, services modules repository, and satellite data catalogues.

The computing facility, which is the actual component of the G-POD environment that executes the processing, is composed by several logical elements:

- A single Computing Element (CE), which acts as a master node in terms of job scheduling policy control;
- Several Worker Nodes (WNs), which provide the required computation capability;
- The Storage Elements (SE), where the overall archived data are stored.

The computing resources dedicated to the P-SBAS service are described in Table 5.1 They are managed by grid and cloud technologies running on Linux operating systems. In addition, a number of utility software such as IDL, Matlab, Sentinel Toolbox, NEST and Gamma are also available [27]. Moreover, thanks to the architecture flexibility, G-POD can easily federate additional computing and storage resources exceeding in this way its base processing capacity. Note also that these G-POD nominal resources belong to ESRIN-ESA but, recently, a new CE has been installed and federated at the IREA-CNR premises [28], which controls 15 WNs (see Table 5.1) fully dedicated to the P-SBAS service.

Table 5.1. G-POD resources dedicated to the P-SBAS web tool

	ESRIN-ESA	**IREA-CNR**
Worker Nodes	15	15
Virtual CPU	60	30
CPU Type	Intel(R) Xeon(R) X5650 2.67 GHz	Intel(R) Xeon(R) E5-2680 2.70 GHz
RAM /WN	32	64
Storage	2TB	4TB
Network	1 Gbps	1 Gbps

5.2 The P-SBAS Web Tool within the G-POD Environment

In this section, we first present the main steps to be accomplished to implement the P-SBAS web tool within the on-demand environment of G-POD; subsequently, the key features and characteristics of the implemented P-SBAS web tool, accessible at [29], are described.

5.2.1 P-SBAS Web Tool: Implementation

The deployment of the unsupervised P-SBAS algorithm, and in general of a scientific application, within the G-POD environment for developing an on-demand tool accessible via the web requires:

- A preliminary study of the algorithm aimed at identifying the main independent processes in which it can be decomposed in order to design the proper application workflow;
- The implementation of the defined workflow in Linux Bash (LB) program language according to the tasks scheduling policy adopted within the G-POD environment. As a result, a set of Linux Bash scripts manages the entire P-SBAS algorithm by properly scheduling the different processing steps and, for each parallel step, by correctly distributing the concurrent jobs among the available WNs;
- The implementation of the Linux Bash scripts for executing the processing steps that constitute the P-SBAS workflow and that are launched on the different WNs;
- The definition and implementation of the web user interface, which allows the configuration and selection of input parameters, thresholds, processing settings, etc.;
- The identification of the whole processing outputs and the implementation of appropriate procedures to collect and make them available for download at the end of the DInSAR processing.

5.2.2 P-SBAS Web Tool: Features and Characteristics

The access to the developed on-demand P-SBAS web tool on GPOD requires a mandatory registration step on the ESA web portal [30], by creating an ESA Earth Observation Single Sign-On (EO-SSO) account. Then, such an account has to be activated to access the P-SBAS web tool via a request to the Operation Support Team (eo-gpod@esa.int). Following this preliminary step, the use of the P-SBAS web tool within G-POD is rather straightforward because it simply requires the selection, from the ESA archives, of SAR data to be exploited, and the setting of few parameters needed for the multi-temporal DInSAR processing.

The P-SBAS Service is indeed provided with a user-friendly interface, see Figure 5.2, which allows any user to produce multi-temporal DInSAR products. In particular, the user has to carry out, as only task, the following few actions:

- Select the computing facilities/resources among those available through the platform, insert the interferometric task caption, select where the final results will be made available to the user, and set the task priority (Figure 5.2-a);
- Select the AoI to be processed and the DInSAR reference pixel. The AoI selection is made via a bounding box over an interactive geographical map client (Figure 5.2-b). Instead, the reference point selection is carried out by shifting a place mark over the map client. It is anyway possible to insert geographic coordinates of a known point in the appropriate fields. As a general hint for the reference point selection, it is strongly suggested that it will be located in a stable and (expected) coherent area;
- Indicate the temporal span used for the subsequent SAR raw data catalogue querying (Figure 5.2-c);

- Select the ESA data catalogue from which the SAR acquisitions are automatically retrieved. The Virtual Archive 4 (VA4) catalogue [22] is the main source of SAR data for the P-SBAS web tool. Moreover the possibility to select data by relative orbit number is also provided (Figure 5.2-d);
- Optionally, enter some basic interferometric SAR parameters such as: spatial and temporal baseline thresholds, approximate size (in meters) of the resulting ground pixel, filtering coefficient and noise thresholds (Figure 5.2-e). It is strongly suggested that users skilled on DInSAR processing perform the setting of these parameters. However, to simplify this task for less expert users, default values for each parameter are also provided. It is worth highlighting that these default values are the result of extensive and long-term analyses of ESA C-band data and are suitable for a large part of case studies.

Note that a flag is also available to limit the DInSAR analysis to the generation of multi-temporal interferograms only.

Once the actions above discussed have been performed, the user can start the run of the P-SBAS processing chain on the allocated computing facilities. While running, the user has also the possibility to monitor the process status (Figure 5.2-f) and, once the job is terminated, the final results are available for download from either the G-POD web portal or a user-defined FTP server.

5.3 Experimental Results

An extensive test and validation activity has been carried out in the six-month period from January to June 2015 in order to assess the performance of the P-SBAS processing chain implemented within the

Figure 5.2. P-SBAS tool web portal. (a) Main processing Parameters window, (b) geographic map client for the AoI and data selection, (c) investigated time interval selection, (d) SAR data catalogue query, (e) interferometric processing parameters window, and (f) job status monitoring.

G-POD environment and accessible via the web. To this aim, several datasets, acquired in the 2002–2010 time period by the ENVISAT sensor, have been processed. Figure 5.3 pictorially shows this intensive P-SBAS service testing activity; moreover, as an example, the computed mean deformation velocity maps relevant to three representative test cases are also shown. In particular, the selected areas are characterized by very heterogeneous behaviors in terms of surface deformation phenomena and interferometric noise sources. Indeed, the Mt. Etna volcano (Figure 5.3-b) is characterized by a complex deformation behavior interested by several eruptions, fault movement and water extraction.

In particular are clearly visible the deformations associated to the activities along the Trecastagni-Mascalucia and Pernicana faults. Figure 5.3-c, instead, presents the displacement map relative to the city of Roma, which is interested by significant subsidence along the Tevere River, due to the compaction of the alluvial deposits.

Finally, in Figure 5.3-d, it is presented the mean deformation velocity map computed over the Tokyo city area, a wide urbanized region affected by subsidence phenomena.

Overall, the whole test and validation activity involved 30 datasets and the achieved results are summarized in Table 5.2. Note that the exploited datasets consist of SAR data stacks ranging from 15 to 73 acquisitions per area (the average corresponds to 38 SAR acquisitions), which lead to the generation of interferogram sequences spanning the range from 33 to 221 (the average corresponds to 105 interferograms). The spatial extent of each investigated area is of about 100x100 km, which corresponds to about 30000 × 5000 full resolution pixels along azimuth and range directions, respectively. Moreover, a spatial averaging (multilook) of 20 pixels along azimuth and four pixels along range is performed within the interferogram generation step for noise reduction purposes.

The computing resources allocated for each processed dataset consisted in 4 G-POD working nodes, whose characteristics are described in Table 5.1. Moreover, it is worth noting that such processing resources can be further expanded whether requested; as a matter of fact, multiple runs can be executed

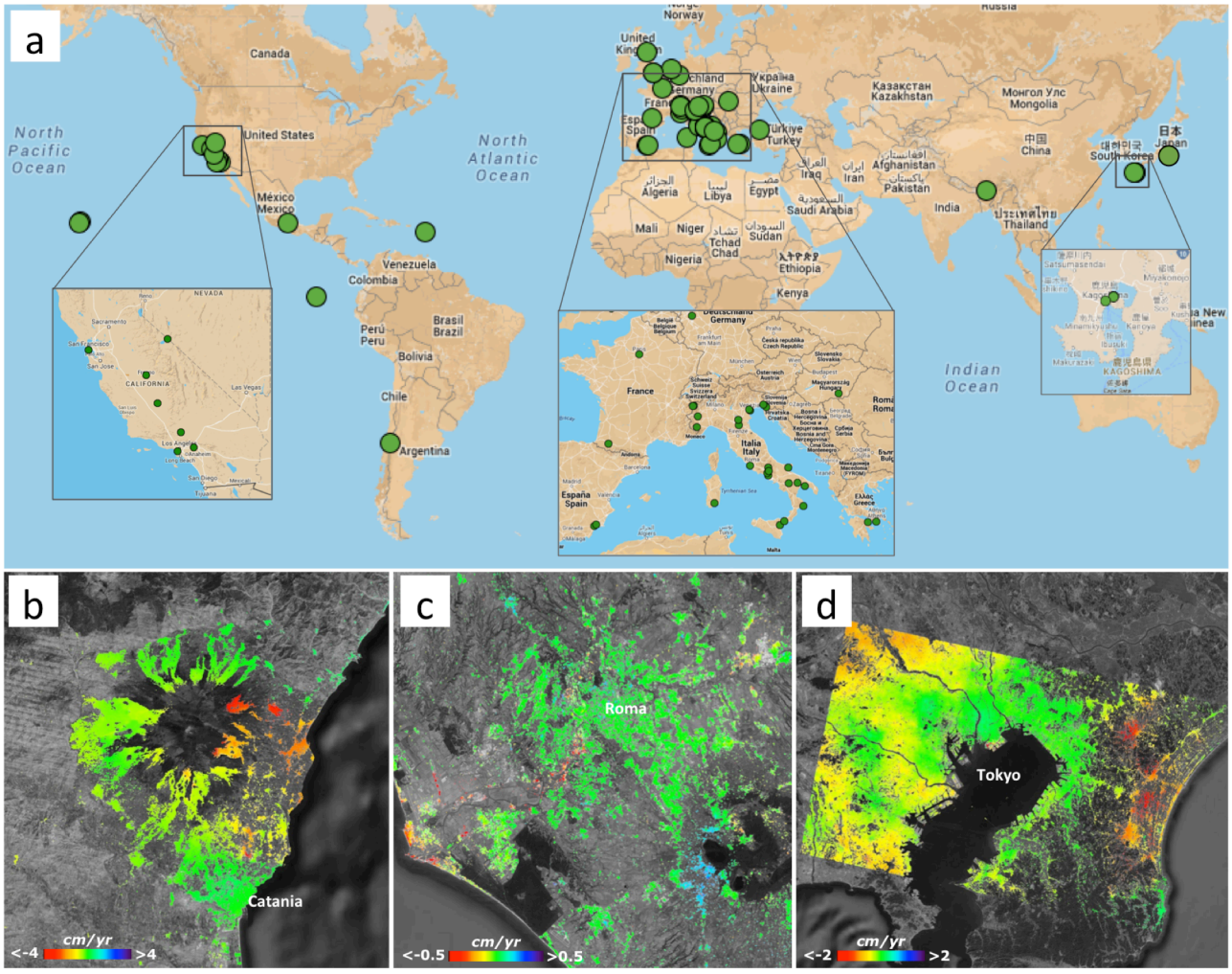


Figure 5.3: P-SBAS processing status. (a) Pictorial representation of the processing results achieved by the users through the P-SBAS web tool within G-POD, during the 6-month period January–June 2015, (b), (c), and (d) are the retrieved mean deformation velocity maps relevant to the Mt. Etna (Italy), Roma (Italy) and Tokyo (Japan) areas, respectively.

in parallel if needed, by scaling up the selected base infrastructure and plugging-in additional external resources (e.g., cloud computing). According to the exploited G-POD hardware configuration and the investigated SAR datasets, the full P-SBAS processing requires an average time of slightly less than one day (0.94 days), as shown in Table 5.2.

In addition it is worth remarking that the implemented unsupervised P-SBAS solution does not significantly affect the quality of the retrieved DInSAR results, in terms of the accuracies of the retrieved measurements. To verify this issue we have compared the achieved P-SBAS results with

those obtained through GPS measurements. We present in the following the results relevant to the ENVISAT datasets available over three test sites: the Campi Flegrei caldera (Italy), the Los Angeles area (US) and the Mauna Loa and Kilauea volcanoes in Hawaii (US). These areas are interested by different deformation behaviors; indeed, the Campi Flegrei is a well-known active volcanic zone close to the Napoli city (Italy), which is experiencing an almost continuous uplift of the Earth surface since late 2004 [31]. Los Angeles is a wide urbanized region affected by oil extraction, aquifer exploitation (implying periodic displacements) and seismic activity [32]. Finally, Hawaii volcanoes are among the most active in the world, as testified by the frequent unrest episodes and eruptions [33] and thus implying a strongly non-linear deformation behavior of the ground.

Table 5.2. ENVISAT SAR data sets processed through the P-SBAS web tool during the 6-month testing period January-June 2015.

Site*	Images	Interf	Coherent pixels	Elapsed time [day]
Tokyo	29	86	407600	0.66
Los Angeles	38	109	967562	1.68
Hawaii_471	37	106	1105564	1.41
Hawaii_200	44	122	536145	0.86
Hawaii_408	36	105	544544	0.80
Hawaii_343	20	50	405365	0.52
Hawaii_136	15	34	448017	0.50
Hawaii_114	19	33	681393	0.44
S. Francisco	43	126	282653	0.92

Galápagos	63	187	617029	2.11
London	49	138	282496	0.87
Homs_307	34	93	288442	0.60
Homs_78	33	94	136727	0.54
Long Valley	50	131	619025	1.99
Roma	39	112	126059	0.62
Ferrara	20	51	113850	0.74
Paris	50	148	253886	1.02
Atene	44	76	124758	0.48
Amsterdam	73	221	143743	1.13
Messico City	50	140	284014	0.97
Racconigi	34	92	157682	0.96
Napoli_36	57	168	119159	0.85
Fresno	32	91	193371	0.63
Lorca	27	75	576065	1.48
Sacramento	42	118	130383	0.89
Copahue_261	31	92	133150	0.47
Atacama	18	46	1645812	1.24
Turkey	29	81	115342	0.71
Delano	34	98	193977	1.48
Average	**37.59**	**104.32**	**401165.96**	**0.95**

* Note that the site names have been defined by the users.

Exploited ENVISAT datasets span the 2003–2010 time interval and are composed by 65 (Track 129), 45 (Track 120) and 42 (Track 200) ASAR data for the Campi Flegrei, Los Angeles and Hawaii sites, respectively. Moreover, extended GPS networks have been installed in all the three selected sites. In particular, the Campi Flegrei network is active since 2000 [34], the Los Angeles one has been completed in 2001 [35], while the first GPS station on Hawaii was activated in 1995 [36]. Within the selected test sites, the GPS measurements have been compared with those obtained through the unsupervised P-SBAS processing. In particular, among several, we identified 100 stations that are both located in correspondence with coherent pixels and guarantee a temporal overlap with the P-SBAS measurements of at least two years. Moreover, since we considered the GPS as our reference, we neglected the stations highly affected by noise (i.e., spikes or average of nominal sigma values larger than 4 mm in LOS). As an example, in Figure 5.4, we show the surface deformation mean velocity map retrieved for the three test sites with highlighted, per each of them, the positions of three representative GPS stations; for the same stations, the displacement time series retrieved by both the GPS measurements and the P-SBAS processing are depicted as well. Plots of Figure 5.4 permit us also to show the capability of the unsupervised P-SBAS algorithm to correctly detect and follow complex deformation behavior characterized by periodic signals and strong non-linearity.

For all the 100 selected stations we calculated the standard deviation of the difference between the P-SBAS and GPS time series, obtaining average values of about 4.0 mm, 6.6 mm and 11.8 mm for the Campi Flegrei, Los Angeles and Hawaii sites, respectively Note that, even if in this case it is assumed as the reference, the GPS measurements are themselves affected by noise. Therefore, per each station of the three test sites, we empirically estimated the standard deviations of the GPS time series with respect to a smoothed version of the time series itself, in the time period common to the SAR acquisitions. Obtained values are about 1 mm, 3 mm and 6 mm, for the Campi Flegrei, Los Angeles and Hawaii test sites, respectively. By considering these values, we may remove from the P-SBAS /

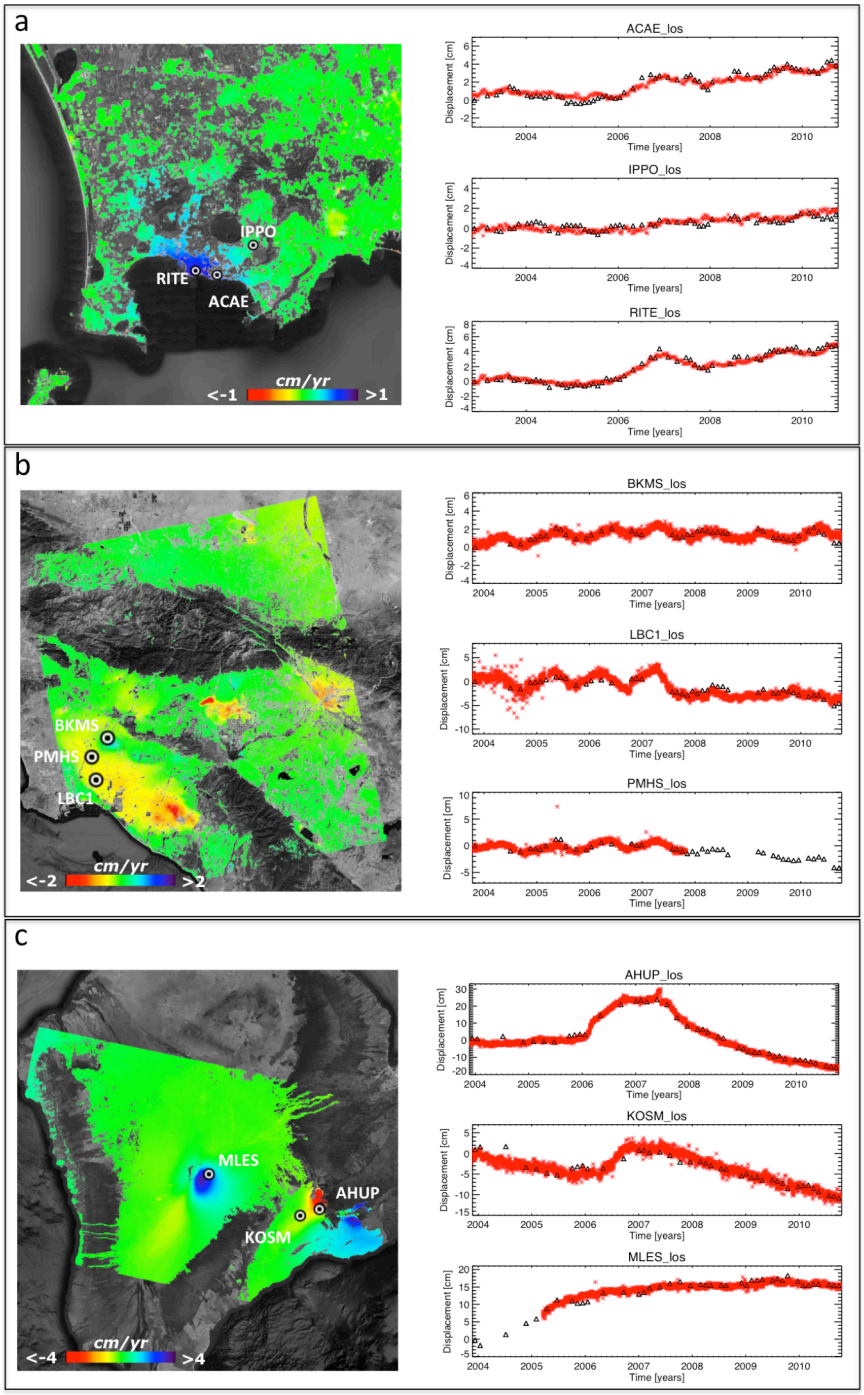


Figure 5.4: Unsupervised P-SBAS processing chain validation experiments. Per each of the three selected test sites are shown: the mean deformation velocity maps retrieved from the P-SBAS analysis; the locations of three selected GPS stations; the comparison between the GPS (red stars) and the P-SBAS (black triangles) surface deformation time series relevant to the stations identified in the corresponding map; the standard deviations (σ) of the difference between the GPS and the DInSAR measurements. (a) Campi Flegrei caldera. (b) Los Angeles. (c) Hawaii.

GPS time series standard deviations the bias due to the estimated errors relevant to the geodetic measurements.

The achieved results very well match with the expected SBAS technique accuracy [23]. Accordingly, these validation experiments confirm the good agreement between the conventional SBAS and the unsupervised P-SBAS versions, demonstrating that the latter preserves the accuracies of the SBAS results.

5.4 Summary

In this chapter we presented an on-demand and unsupervised web tool for the generation of DInSAR deformation time series and mean deformation velocity map. The motivation for the development of such a tool comes from the idea of moving both DInSAR algorithms and computational resources close to SAR data in order to pursue the full exploitation of the existing (and forthcoming) archives. In particular, an operational DInSAR processing chain, accessible through the web, has been set up to process ESA SAR data by properly exploiting (a) the P-SBAS multi-temporal DInSAR algorithm, (b) the high-performance and sizeable computing resources provided by the G-POD environment, and (c) the directly accessible ESA archives. Accordingly, the implemented tool lets the users free from the necessity of ordering and downloading SAR data, as well as of procuring a DInSAR processing algorithm and the needed computing resources.

The proposed solution allows the provisioning of a fully-unsupervised P-SBAS processing service to non-expert users, as well as to expert ones that can benefit from a tool that permit a fast exploration of a study area and a better control of the processing chain (e.g., through the interactive interferogram selection), thus expanding the access to high quality multi-temporal DInSAR results. Such a service provides the users not only with a fully-functional on-demand P-SBAS web tool but also with a

technical and scientific support for either addressing possible issues or suggesting optimal usage. Such a support can be obtained through the dedicated G-POD team (eo-gpod@esa.int); moreover, a detailed user manual is also available for download [37] to allow users to exploit the tool as easily as possible. Furthermore, the flexibility of the exploited G-POD architecture makes simple to federate additional computing and storage resources which can be also provided through cloud environments. Indeed, conjugating this approach with cloud computing infrastructures is foreseen because it will permit to avoid possible bottlenecks, represented by the intrinsic limited resources of in-house facilities [27], and to extend this service to a wider number of users.

In addition, the availability of this and other web tools within the ESA's GEP initiative, will allow users to remotely process the data, as well as to share, promote, and discover the obtained results among the platform community, thus fostering the creation of an environment where doing science in a new way.

Finally, it is worth remarking the coming possibility, by the end of 2015, to dispose of a similar web tool able to process Sentinel-1 TOPS data, thus allowing users to generate DInSAR results at theoretically Worldwide scale.

References

[1] Massonnet, D.; Feigl, K.L. Radar interferometry and its application to changes in the Earth's surface. Rev. Geophys. 1998, 36, 441–500

[2] Burgmann, R.; Rosen P.A.; Fielding E.J. Synthetic aperture radar interferometry to measure Earth's surface topography and its deformation. Ann. Rev. Earth Planet. Sci. 2000, 28, 169–209.

[3] Gabriel, A.K.; Goldstein, R.M.; Zebker H.A. Mapping small elevation changes over large areas: Differential interferometry. J. Geophys. Res. 1989, 94, 9183–9191.

[4] Franceschetti G.; Lanari R. Synthetic Aperture Radar Processing; CRC Press: Boca Raton, FL, USA, 1999.

[5] Massonnet, D.; Rossi, M.; Carmona, C.; Adragna, F.; Peltzer, G.; Feigl K.; Rabaute, T. The displacement field of the Landers earthquake mapped by radar interferometry. Nature 1993, 364, 138–142.

[6] Ferretti, A.; Prati, C.; Rocca, F. Permanent scatterers in SAR interferometry. IEEE Trans. Geosci. Remote Sens. 2001, 39, 8–20.

[7] Werner C.; Wegmüller U.; Strozzi T.; Wiesmann A. Interferometric point target analysis for deformation mapping. In Proceedings of the Geoscience and Remote Sensing Symposium, Toulouse, France, 21–25 July 2003; pp. 4362–4364.

[8] Hooper, A.; Zebker, H.; Segall, P.; Kampes, B. A new method for measuring deformation on volcanoes and other natural terrains using InSAR persistent scatterers. Geophys. Res. Lett. 2004, 31, L23611.

[9] Kampes, B.M. Radar Interferometry: Persistent Scatterer Technique; Springer-Verlag: Dordrecht, The Netherlands, 2006.

[10] Berardino, P.; Fornaro, G.; Lanari, R.; Sansosti, E. A new algorithm for surface deformation monitoring based on small baseline differential SAR interferograms. IEEE Trans. Geosci. Remote Sens. 2002, 40, 2375–2383.

[11] Mora, O.; Mallorquí, J.J.; Broquetas, A. Linear and nonlinear terrain deformation maps from a reduced set of interferometric SAR images. IEEE Trans. Geosci. Remote Sens. 2003, 41, 2243–2253.

[12] Crosetto, M.; Crippa, B.; Biescas, E. Early detection and in-depth analysis of deformation phenomena by radar interferometry. Eng. Geol. 2005, 79, 81–91.

[13] Sansosti,E.; Casu,F.; Manzo,M.; Lanari, R.Space-borne radar interferometry techniques for the generation of deformation time series: An advanced tool for Earth's surface displacement analysis. Geophys. Res. Lett. 2010, 37, L20305.

[14] Lanari, R.; Mora, O.; Manunta, M.; Mallorquí, J.J.; Berardino, P.; Sansosti, E. A small-baseline approach for investigating deformations on full-resolution differential SAR interferograms. Int. IEEE Trans. Geosci. Remote Sens. 2004, 42, 1377–1386.

[15] Strang,G.LinearAlgebraandItsApplications;HarcourtBraceJovanovich:Orlando,FL,USA,1988.

[16] Zebker, H.A.; Villasenor, J. Decorrelation in interferometric radar echoes. IEEE Trans. Geosci. Remote Sens. 1992, 30, 950–959.

[17] Casu, F.; Elefante, S.; Imperatore, P.; Zinno, I.; Manunta, M.; De Luca, C.; Lanari, R. SBAS-DInSAR parallel processing for deformation time-series computation. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2014, 7, 3285–3296.

[18] Zinno, I.; Mossucca, L.; Elefante, S.; Luca, C.; Casola V.; Terzo O.; Casu F.; Lanari R. Cloud computing for earth surface deformation analysis via spaceborne radar imaging: A case study. IEEE Trans. Cloud Comput. 2015, 99, doi:10.1109/TCC.2015.2440267.

[19] Rosen, P.A.; Hensley, S.; Peltzer, G.; Simons, M. Updated repeat orbit interferometry package released. EOS Trans. Am. Geophys. Union 2004, 85, 47.

[20] Agram,P.S.; Jolivet,R.; Riel,B.; Lin,Y.N.; Simons,M.; Hetland,E.; Doin,M.P.; Lassere,C. *New radar interferometric time series analysis toolbox released*. Eos Trans. AGU 2013, 94, 69–70.

[21] Marchetti,P.G.; Rivolta,G.; D'elia,S.; Farres,J.; Gobron,N.; Mason,G. "A Model for the Scientific Exploitation of Earth Observation Missions": The ESA Research and Service Support. IEEE Geosci. Newsl. 2012, 162, 10–18.

[22] VirtualArchive4.Availableonline:http://eo-virtual-archive4.esa.int/(accessedon9September2015).

[23] Lanari, R.; Casu, F.; Manzo, M.; Zeni, G.; Berardino, P.; Manunta, M.; Pepe, A. An overview of the small baseline subset algorithm: A DInSAR technique for surface deformation analysis. Pure Appl. Geophys. 2007, 164, 637–661.

[24] Geohazard Exploitation Platform. Available online: https://geohazards-tep.eo.esa.int/#! (accessed on 9 September 2015).

[25] Geohazard Supersites and Natural Laboratories. Available online: http://supersites. earthobservations.org (accessed on 9 September 2015).

[26] CEOS. Available online: http://www.ceos.org/index.php?option=com_content&view=category&layout=blog&id=355&Itemid=491 (accessed on 9 September 2015).

[27] GRID Processing on Demand. Available online: http://wiki.services.eoportal.org/ tiki-index.php?page=GPOD% 20Wiki (accessed on 9 September 2015).

[28] IREA. Available on line :http://www.irea.cnr.it/en/index.php(accessedon9September2015).

[29] G-POD. Available on line only for registered users: http://gpod.eo.esa.int/services/SBAS(accessed on 30 October 2015).

[30] European Space Agency. Available online: https://earth.esa.int/web/guest/missions/ esa-operational-eo-missions (accessed on 9 September 2015).

[31] Trasatti,E.; Casu,F.; Giunchi,C.; Pepe,S.; Solaro,G.; Tagliaventi,S.; Berardino,P.; Manzo,M.; Pepe, A.; Ricciardi, G.P.; et al. The 2004–2006 uplift episode at Campi Flegrei caldera (Italy): Constraints from SBAS-DInSAR ENVISAT data and Bayesian source inference. Geophys. Res. Lett. 2008, 35, L073078.

[32] Lanari, R.; Lundgren, P.; Manzo, M.; Casu, F. Satellite radar interferometry time series analysis of surface deformation for Los Angeles, California. Geophys. Res. Lett. 2004, 31, L23613.

[33] Poland, M.P.; Sutton, A.J.; Gerlach, T.M. Magma degassing triggered by static decompression at Kīlauea Volcano, Hawaii. Geophys. Res. Lett. 2009, 36, L16306.

[34] DeMartino,P.; Tammaro,U.; Obrizzo,F. GPS time series at Campi Flegrei caldera (2000–2013). Ann. Geophys. 2014, 57, S0213.

[35] SCIGN. Available online: http://www.scign.com.

[36] UNAVCO.Availableonline:http://www.unavco.org/data/gps-gnss/data-access-methods/dai2/app/dai2.html#scope=All;boundingBox=16.4296,-162.2705,23.2049,-151.7236 (accessed on 10 November 2015).

[37] P-SBAS Manual. Available online: https://wiki.services.eoportal.org/tiki-download_file.php?fileId=883.

Conclusion

The evolution of the EO scenario during the last decades has been characterized by the constant increase of data archives at disposal of the scientific community. However, the development of effective solutions able to properly deal with the storage, the accessibility, the transfer and above all, the processing of such a huge data flow is strongly needed.

In this work we investigated the issues related to efficiently and effectively process the ever increasing amount of satellite data in the Earth Observation field, more specifically through the well-known DInSAR SBAS technique, which allows generating displacement time series of the Earth surface by using spaceborne SAR acquisitions. In particular, we concentrated on the use of the SBAS algorithm within CC infrastructures to investigate its effectiveness for EO applications.

To this aim, we first developed a new SBAS solution able to benefit from the multi-node and multi-core HPC architectures. The developed parallel SBAS solution (P-SBAS) has been tested and validated in an in-house cluster, demonstrating that it is particularly suitable to effectively exploit the parallelism offered by dedicated computational platforms, since it presents a deviation from the maximum achievable speed-up of 17% by using 16 nodes.

Second, the proposed P-SBAS solution has also been tested in cloud-computing environment. Different cloud architectural solutions have been implemented to identify the better configuration for the generation of very large-scale ground deformation analysis. In particular, we initially developed an architecture based on a centralized NFS storage (CS), which addressed the P-SBAS data dependency in a very straightforward way. However, the performed experimental analysis highlighted that the CS P-SBAS solution, which entails a complex communication pattern among the computing nodes, need specific strategies to minimize the data transfer between different nodes and, as a consequence, to avoid the network and I/O saturation, that can limit the overall algorithm scalability. Accordingly, we

subsequently developed a distributed NFS storage architecture (DS) specifically designed to minimize the I/O and data transfer workload among the different computing nodes in order to achieve good scalable performances without necessarily requiring very high network and I/O performance resources . The experimental results carried out within the AWS EC2 infrastructure showed that the distributed storage configuration presents very good performances in terms of processing elapsed times when the size of data-set and the number of exploited nodes increases. The DS solution exhibits a very good scalability by exploiting up to 32 AWS instances, with a noticeable improvement with respect to the CS solution. In particular, with 32 AWS instances, the DS solution permits to achive a processing time reduction of around the 21% with respect to the CS solution, implying also a processing cost reduction of around 17%.

Moreover, by benefiting from the exploited public cloud infrastructure, an assessment analysis of the P-SBAS processing costs, depending on the type and number of the exploited AWS cloud resources, has been accomplished, thus permitting us to evaluate the most suitable configuration in accordance with specific time/cost requirements.

The proposed P-SBAS solution allowed us also to set up an on-demand and unsupervised web tool for the generation of DInSAR deformation time series. Such a tool, integrated in the ESA G-POD environment, is accessible and operative for the scientific community and provides a direct access to the ESA data archives and to high performance computing resources. The development of such a tool is of a great benefit for the EO community represented not only by research institutions, but also by governmental bodies and policy makers that are much more interested on results than on processing.

Summarizing in this work, we highlighted that cloud computing represents a valid tool for complex scientific applications; indeed, the performed experimental analysis and the result obtained through the P-SBAS cloud implementation proved that cloud technologies result to be absolutely competitive in terms of achieved scalable performance with respect to in-house HPC cluster solution. The high

customization level and the on-demand services of cloud computing infrastructures are crucial characteristics to make demanding application, as the P-SBAS one, more flexible and user friendly. Moreover, the extensive availability of cloud computing resources perfectly matches with the perspective of processing, also on a continuous basis, the huge data flow which is going to be provided by the Sentinel-1 satellite constellation.

The joint exploitation of the P-SBAS algoritm, cloud architectures and unsupervised processing, opens new intriguing perspectives in the Earth Observation field and, particularly within DInSAR, permits to move ground deformation studies from regional to nearly continental scale. Finally, the availability of advanced DInSAR products, in reasonable time frames and at sustainable costs, has important consequence not only in scientific research fields, but also in civil protection context and institutional services. Indeed, Earth surface deformation detection and monitoring play a key role in defining appropriate strategies for the assessment and the mitigation of risks due to both natural and man-made hazards.

Overview on Distributed Computing Paradigms

Scientific computing involves the construction of mathematical models and numerical solution technique to solve scientific and engineering problems. These models often require a huge number of computing resources to perform large-scale experiments or to cut down the computational complexity into a reasonable time frame.

These needs have been initially addressed with dedicated high-performance computing (HPC) infrastructures such as clusters or with a pool of networked machines in the same department. With the advent of Grid computing [1] new opportunities became available to scientists, the Grid computing could offer on demand the power required to perform large experiments, by relying on a network of machines, potentially extended all over the world. The use of Grids for scientific computing [2] has become so successful that many international projects led to the establishment of worldwide infrastructures available for computational science. An example is the Enabling Grid for E-SciencE (EGEE) [3] is an initiative funded by European Commission that connects more than 91 institutions in Europe, Asia, and United States of America, to construct the largest multi-science computing Grid infrastructure of the world.

The current emerging trend in delivering IT services is the Cloud Computing. The Cloud Computing offers to the end users a variety of services covering the entire computing stack, from the hardware to the application level, by charging them on a *pay per use* basis. Another important feature, from which scientist can benefit, is the ability to scale up and down the computing infrastructure according to the application requirements and the budget of the users. By using cloud-based technologies, scientist can

have easy access to large distributed infrastructures and completely customize their execution environment, thus providing the perfect setup for their experiments. Moreover, by renting the infrastructure on a *pay per use* basis, they can have immediate access to required resources without any capacity planning and they are free to release them when resources are no longer needed.

The topic of this chapter is to provide to the reader an overview on three different High Performance Computing (HPC) model, extremely suitable for scientific application as Earth Observation (EO) application. In particular we achieve a high level description of cluster computing, grid computing and cloud computing models, highlighting the principal advantages and disadvantages of each computing solution.

A.1 Cluster Computing

A cluster is a collection of a parallel or distributed computers witch are interconnected among themselves using high-speed networks, such as gigabit Ethernet or Infiniband. They work together in the execution of compute intensive and data intensive tasks that would be not feasible to execute on a single computer. Cluster is used mainly for high availability, load-balancing and for compute purpose [4]. They are used for high availability purpose as they maintain redundant nodes, which are used to provide service when system components fail. The performance of the system is improved here because even if one node fails there is another nodes which could carry the task and eliminates the single point of failure. So users always have some components to work with even in the presence of fault.

When multiple computers are linked together in a cluster, they share computational workload as a single virtual computer. From the user view point they are multiple machines, but they function as a single virtual machine.

In the following we describe some advantages and disadvantages of cluster computing architecture in accordance with [5].

Advantages of Cluster Computing:

1) *Manageability:* It takes a lot of effort, cost and money to manage a large number of components. But, with cluster, large numbers of components are combined to work as a single entity. So, management becomes easy.
2) *Single System Image:* Again, with cluster, user just gets the feel that he is working with a single system, but actually he is working with a large number of components. He need not worry about that components, he only needs to manage a single system image.
3) *High Availability:* As all the components are replicas of each other, so if one component goes down because of any technical reason, then some other component can takes its place, and user can continue to work with the system.

Disadvantages of Cluster Computing:

1) *Programmability Issues:* This might be the case if the components are different in terms of software from each other, and then there may be issues when combining all of them together as a single entity.
2) *Problem in Finding Fault:* Because we are dealing with a single entity, so problem may arise when finding out fault that which of the component has some problem associated with it.
3) *Difficult to handle by a Layman:* As cluster computing involves merging different or same components together with different programmability, so a non-professional person may find it difficult to manage.

A.2 Grid Computing

Grid computing [6] combines computers from multiple administrative domains to reach a common goal, to solve a single task. The Grid solution uses multiple clusters that are coupled, heterogeneous and are geographically dispersed. The individual user gets access to the resources (like processors, storage, data, etc..) on demand with little or no knowledge of the fact that where those resources are physically located [7].

Buyya et. al. [8] defined grid as a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, capability, performance, cost and users quality-of-service requirements. In the following we describe some advantages and disadvantages of grid computing architecture in accordance with [5].

Advantages of Grid Computing:

1) *Access to Additional Resources*: In addition to CPU and other storage resources, a grid can also provide other resources as well.
2) *Resource Balancing*: A grid incorporates large number of systems into a single system image. For applications that are grid enabled, grid performs the resource balancing by scheduling grid jobs on machines that are showing low utilization.
3) *Reliability*: The systems in grid are cheap and geographically dispersed. If, for example, there is power or cooling failure at one site, then that will not affect the other site, thus high reliability will be there specially in case of real time systems

Disadvantages of Grid Computing

1) *Not Stable*: Grid software and standards are not stable in comparison to other computing. Its standards are still evolving.
2) *High Internet Connection Required*: Gathering and assembling various resources from geographically dispersed sites require high internet connection which results in high monetary cost.
3) *Different Administrator Domains*: Sometimes political issues arise when sharing resources among different domains. Some additional tools are required for having proper syncing and managing among different environments.

A.3 Cloud Computing

Cloud computing is the new computing paradigm, which provides large pool of dynamical scalable and virtual resources as a service on demand. The main principle behind cloud computing model is to offer computing, storage, and software as a service or as a utility. We just need internet to use these utilities. Buyya et al. (2009) [8] have defined it as follows: "Cloud is a parallel and distributed computing system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements (SLA) established through negotiation between the service provider and consumers." Cloud infrastructure includes scalable resources in storage, network, and compute. It also contain virtualized infrastructure and provide these services to the users over Internet.

Cloud computing cuts the operational and capital costs and allow the IT departments to focus on strategic projects instead of keeping the datacenter running. It provides the services on Infrastructure level, Platform level, and Software level. It provides many features such as speed, scalability of resources, parallel processing, just pay the used resources, choose another technology at any time to

further work, 24/7 availability of services, device and location independent, provides reliability and security etc. Cloud has five essential features such as rapid elasticity, measured services, on-demand self-service, resource pooling, and board network access.

Cloud computing provides basically three kinds of service: Software as a Service (*SaaS*), Platform as a Service (*PaaS*) and Infrastructure as a Service (*IaaS*.)

- *SaaS* is a kind of services where in many users can make use of the software hosted by the service provider and pay only for time its being used. It will be better than buying the hardware and software as keeps off the burden of updating the software to the latest version, licensing and is of course more economical [4].
- *PaaS* provides a high-level integrated environment to design, build, test, deploy and update online custom applications. Some example service providers are Google's App Engine, Microsoft Azure.
- *IaaS* refers to the services provided to the users to use processing power, storage, network and other computing resources, to run any software including operating systems and applications. Amazon is one of the major players in providing IaaS solutions. Amazon Elastic Compute Cloud (EC2) provides a large computing infrastructure and a service based on hardware virtualization. By using Amazon Web Services, users can create Amazon Machine Images (AMIs) and save them as templates from which multiple instances can be run. It is possible to run either Windows or Linux virtual machines and the user is charged per hour for each of the instances running. Amazon also provides storage services with the Amazon Simple Storage Service (S3), users can use Amazon S3 to host large amount of data accessible from anywhere.

The Cloud infrastructure delivers its service through four deployment models namely, public cloud, private cloud, community cloud and hybrid cloud, in detail:

Public clouds in this deployment the cloud infrastructure is accessible to general public and shared in a *pay as you go* model of payment. The cloud resources are accessible via the internet and the provider is responsible for ensuring the economies of scale and the management of the shared infrastructure. In this model clients can choose security level they need, and negotiate for service levels. Amazon Web Services EC2 is a public cloud.

Private clouds are another deployment model for cloud services. In this model the cloud resources are not shared by unknown third parties. The cloud resources in this model may be located within the client organization premises or offsite. In this model the client security and compliance requirements are not affected though this offering does not bring the benefits associated with reduced capital expenditure in IT infrastructure investments. In this type of cloud the general public does not have access to the private cloud neither does the organization use the public cloud.

Hybrid clouds as its name implies is a model of deployment which combines different clouds for example the private and public clouds. In this model the combined clouds retains their identities but are bound together by standardized technology. In this type of cloud the general public does not have access to the cloud, but the organization uses infrastructure in both the public and private cloud.

Community clouds are the fourth deployment model that can be used to deliver cloud computing services. In this model the cloud infrastructure is shared by multiple organizations or institutions that have a shared concern or interest such as compliance considerations, security requirements. This type of cloud may be managed by the organization or by a third party and may be located on- premises or

off-premises. In this type of cloud both the public and the organizations forming the community cloud have access to the cloud services offered by the community cloud.

In the following we describe some advantages and disadvantages of cloud computing architecture in accordance with [5].

Advantages of Cloud Computing

1) *Shared Resources*: Cloud computing share resources to provide the services to multiple users. That's why it can easily provide the facility like scale up and scale down the resources on demand.
2) *Pay-As-You-Go*: Users just need to pay only for those resources which are used by them. They can demand for more resources if they required latter on and they can also release their resources after use.
3) *Better Hardware Management*: It is easy for cloud service provider to manage the hardware easily because all computers run the same hardware [9].

Disadvantages of Cloud Computing

1) *Less Reliability*: Cloud Computing is less reliable because it used to share the resources with multiple users. So there is possibility to steal the data of a user or data of one organization may mix with the data of another organization.
2) *Internet*: The main requirement for users to use the services of cloud computing is internet. Users required high speed of internet connection [10]. Unavailability of internet would cause unavailability of data.

3) *Non-Interoperability*: If user stored data in one cloud then later on he/she can't move it to another cloud service provider because there is non-interoperability between cloud based systems [10].

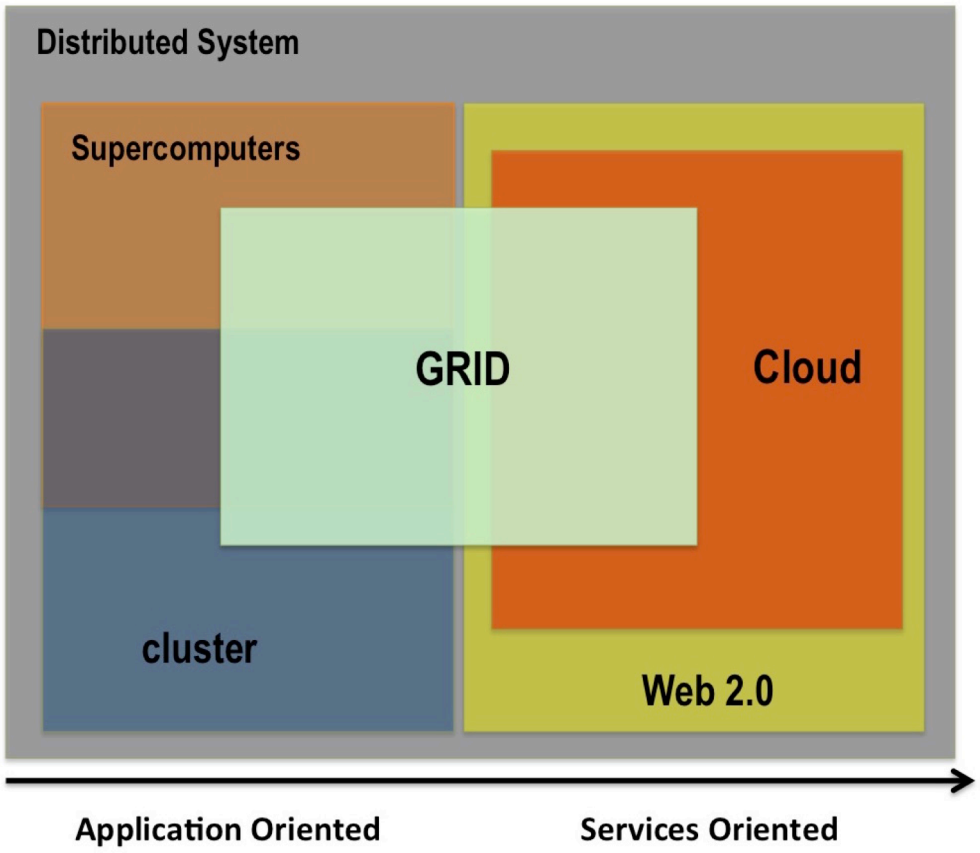


Figure A.1: Relationship of different Distributed Domains

After this high level description of the cluster, grid and cloud architecture and features, in Figure A.1 we emphasize the relationship between all these domains. Web 2.0 covers almost the whole spectrum of service-oriented applications, where Cloud Computing lies at large scale side. Supercomputing and Cluster computing have been more focused on the traditional non-service applications. Grid computing

overlaps with all these fields where it is generally considered of lesser scale than supercomputers and Clouds.

References

[1] Foster and C. Kesselman, The Grid: blueprint for a new computing infrastructure, I.Foster and C. Kesselman, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., November 1998.

[2] M. J. Chin, S. Harvey, S. Jha, and P. V. Coveney, "Scientific Grid Computing: The First Generation," Computing in Science and Engineering, vol. 7, 2005, . pp. 24–32.

[3] F. Gagliardi, M.E. Begin, "EGEE - Providing a Production Quality Grid for e-science," Local to Global Data Interoperability - Challenges and Technologies, Sardinia, Italy, June, 2005

[4] N. Sadashiv and S. M. D. Kumar, "Cluster, grid and cloud computing: A detailed comparison," *Computer Science & Education (ICCSE), 2011 6th International Conference on*, Singapore, 2011, pp. 477-482. doi: 10.1109/ICCSE.2011.6028683

[5] K.Kaur and A.K. Rai, "A comparative analysis: Grid, cluster and cloud computing," International Journal of Advanced Research in Computer and Communication Engineering, vol.3, no.3, pp. 5730-- 5734, 2014.

[6] K. Krauter, R. Buyya, and M. Maheswaran, "A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing", *Jr. of Software Practice and Experience*, 32,(2), pp. 135-164, 2002.

[7] M.Chetty and R. Buyya, "Weaving Computational Grids: How Anal- ogous Are They with Electrical Grids?", *Computing in Science and Engineering (CiSE)*,4, pp. 61-71, 2002.

[8] Buyya R., Yeo C. S., Venugopal S., Broberg J., and Brandic I. (2009) Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing the 5th utility, Future Generation Computer Systems, 25, pp. 599-616.

[9] Gandotra Indu, Abrol Pawanesh, Gupta Pooja, Uppal Rohit and Singh Sandeep (2011) “Cloud Computing Over Cluster, Grid Computing: a Comparative Analysis”, Journal of Grid and Distributed Computing, pp-01-04

[10] Maria S. Perez. “Grid and Cloud Computing”, Retrieved from http://laurel.datsi.fi.upm.es/_media/docencia/a signaturas/ccg/gridcloud.pdf